VARIANCE-COVARIANCE REGULARIZATION IMPROVES REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Transfer learning plays a key role in advancing machine learning models, yet conventional supervised pretraining often undermines feature transferability by prioritizing features that minimize the pretraining loss. In this work, we adapt a self-supervised learning regularization technique from the VICReg method to supervised learning contexts, introducing Variance-Covariance Regularization (VCReg). This adaptation encourages the network to learn high-variance, low-covariance representations, promoting learning more diverse features. We outline best practices for an efficient implementation of our framework, including applying it to the intermediate representations. Through extensive empirical evaluation, we demonstrate that our method significantly enhances transfer learning for images and videos, achieving state-of-the-art performance across numerous tasks and datasets. VCReg also improves performance in scenarios like long-tail learning and hierarchical classification. Additionally, we show its effectiveness may stem from its success in addressing challenges like gradient starvation and neural collapse. In summary, VCReg offers a universally applicable regularization framework that significantly advances transfer learning and highlights the connection between gradient starvation, neural collapse, and feature transferability.

1 INTRODUCTION

Transfer learning enables models to apply knowledge from one domain to enhance performance in another, particularly when data are scarce or costly to obtain (Pan & Yang, 2010; Weiss et al., 2016; Zhuang et al., 2020; Bommasani et al., 2021). One of the key challenges arises during the supervised pretraining phase. In this phase, models often lack detailed information about the downstream tasks to which they will be applied. Nevertheless, they must aim to capture a broad spectrum of features beneficial across various applications (Bengio, 2012; Caruana, 1997; Yosinski et al., 2014). Without proper regularization techniques, these supervised pretrained models tend to overly focus on features that minimize supervised loss, resulting in limited generalization capabilities and issues such as gradient starvation and neural collapse (Zhang et al., 2016; Neyshabur et al., 2017; Zhang et al., 2021; Pezeshki et al., 2021; Papyan et al., 2020; Shwartz-Ziv, 2022).

To tackle these challenges, we adapt the regularization techniques of the self-supervised VICReg method (Bardes et al., 2021) for the supervised learning paradigm. Our method, termed Variance-Covariance Regularization (VCReg), aims to encourage the learning of representations with high variance and low covariance, thus avoiding the overemphasis on features that merely minimize supervised loss. Instead of simply applying VCReg to the final representation of the network, we explore the most effective ways to incorporate it throughout intermediate representations.

The structure of the paper is as follows: we begin with an introduction of our method, including an outline of a fast implementation strategy designed to minimize computational overhead. Following this, we present a series of experiments aimed at validating the method's efficacy across a wide range of tasks, datasets, and architectures. Subsequently, we conduct analyses on the learned representations to demonstrate VCReg's effectiveness in mitigating common issues in transfer learning, such as neural collapse and gradient starvation.

Our paper makes the following contributions:

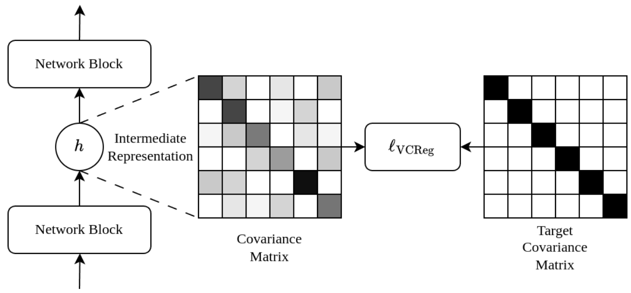


Figure 1: VCReg regularizes the network by encouraging the intermediate representations to have high variance and low covariance. VCReg is applied to the output of each network block to make all the intermediate representations capture diverse features.

1. We introduce a robust strategy for applying VCReg to neural networks, including integrating it into the intermediate layers.
2. We propose a computationally efficient implementation of VCReg. This implementation is optimized to ensure minimal additional computational overhead, allowing for seamless integration into existing workflows.
3. Through extensive experiments on benchmark datasets both in images and videos, we demonstrate that VCReg suppresses the prior state-of-the-art results in transfer learning performance across various network architectures, including ResNet (He et al., 2016), ConvNeXt (Liu et al., 2022), and ViT (Dosovitskiy et al., 2020). Moreover, we also show that VCReg improves performance in diverse scenarios like long-tail learning and hierarchical classification.
4. We investigate the representations learned by VCReg, revealing its effectiveness in combating challenges such as gradient starvation (Pezeshki et al., 2021), neural collapse (Papyan et al., 2020), information compression (Shwartz-Ziv, 2022), and sensitivity to noise.

Before delving into VCReg's details in the following sections, it is key to note its divergence from VICReg, namely by omitting the invariance loss and focusing on variance and covariance loss for a wider application, especially in transfer learning. This approach tackles challenges like gradient starvation and neural collapse, improving neural network training across various architectures. Our work further distinguishes itself by exploring optimal regularization strategies, moving beyond generic application to significantly enhance its effectiveness.

2 Related Work

2.1 Variance-Invariance-Covariance Regularization (VICReg)

VICReg (Bardes et al., 2021) is a novel SSL method that encourages the learned representation to be invariant to data augmentation. However, focusing solely on this invariance criterion can result in the network producing a constant representation, making it invariant to both data augmentation and the input data itself. VICReg primarily regularizes the network by combining variance loss and covariance loss. The variance loss encourages high variance in the learned representations, thereby promoting the learning of diverse features. The covariance loss, on the other hand, aims to minimize redundancy in the learned features by reducing the overlap in information captured by different dimensions of the representation. This dual-objective optimization framework effectively promotes diverse feature learning for SSL (Shwartz-Ziv et al., 2022). To improve the performance of supervised network training, we adapt the SSL feature collapse prevention mechanism from VICReg and propose a variance-covariance regularization method.

To calculate the loss function of VICReg with a batch of data $\{x_1 \dots x_n\}$, we first need to have a pair of inputs (x_i', x_i'') such that x_i' and x_i'' are two augmented versions of the original input x_i. Given the neural network $f_\theta(\cdot)$ and the final representations $z_i' = f_\theta(x_i')$ and $z_i'' = f_\theta(x_i'')$ such that

$z_i', z_i'' \in \mathbb{R}^D$, VICReg minimizes the following loss:

$$\ell_{\mathrm{VICReg}}(z_1' \dots z_n', z_1'' \dots z_n'') = \alpha \ell_{\mathrm{var}}(z_1', \dots, z_n') + \alpha \ell_{\mathrm{var}}(z_1'', \dots, z_n'') + \beta \ell_{\mathrm{cov}}(z_1', \dots, z_n') + \beta \ell_{\mathrm{cov}}(z_1'', \dots, z_n'') + \sum_{i=1}^{n} \ell_{\mathrm{inv}}(z_i', z_i''). \quad (1)$$

The variance and covariance loss functions are respectively defined as:

$$\ell_{\mathrm{var}} = \frac{1}{D} \sum_{i=1}^{D} \max(0, 1 - \sqrt{C_{ii}}), \;\; \ell_{\mathrm{cov}} = \frac{1}{D(D-1)} \sum_{i \neq j} C_{ij}^2 \quad (2)$$

where $C = \frac{1}{N-1} \sum_{i=1}^{N} (z_i - \bar{z})(z_i - \bar{z})^T$ denotes the covariance matrix, and $\bar{z}$ represents the mean vector, given by $\bar{z} = \frac{1}{N} \sum_{i=1}^{N} z_i$.

Building on insights from prior studies (Shwartz-Ziv, 2022; Shwartz-Ziv et al., 2023), it is understood that the invariance term does not play a pivotal role in diversifying features. Consequently, in adapting to the supervised regime, we exclude the invariance term from the regularization.

2.2 Representation Whitening and Feature Diversity Regularizers

Representation whitening is a technique for processing inputs before they enter a network layer. It transforms the input so that its components are uncorrelated with unit variance (Kessy et al., 2018). This transformation achieves enhanced model optimization and generalization. It uses a whitening matrix derived from the data's covariance matrix and results in an identity covariance matrix, thereby aiding gradient flow during training and acting as a lightweight regularizer to reduce overfitting and encourage robust data representations (LeCun et al., 2002).

In addition to whitening as a processing step, additional regularization terms can be introduced to enforce decorrelation in the representations. Various prior works have explored these feature diversity regularization techniques to enhance neural network training (Cogswell et al., 2015; Ayinde et al., 2019; Laakom et al., 2023). These methods encourage diverse features in the representation by adding a regularization term. Recent methods like WLD-Reg (Laakom et al., 2023) and DeCov (Cogswell et al., 2015) also employ covariance-matrix-based regularization to promote feature diversity, similarly to our approach.

However, the studies above mainly focus on the benefits of optimization and generalization for the source task, often neglecting their implications for supervised transfer learning. VCReg distinguishes itself by explicitly targeting enhancements in transfer learning performance. Our results indicate that such regularization techniques yield only modest performance improvements in in-domain evaluations.

2.3 Gradient Starvation and Neural Collapse

Gradient starvation and neural collapse are two recently recognized phenomena that can significantly affect the quality of learned representations and a network's generalization ability (Pezeshki et al., 2021; Papyan et al., 2020; Ben-Shaul et al., 2023). Gradient starvation occurs when certain parameters in a deep learning model receive very small gradients during the training process, thereby leading to slower or non-existent learning for these parameters (Pezeshki et al., 2021). Neural collapse, on the other hand, is a phenomenon observed during the late stages of training when the internal representations of the network tend to collapse towards each other, resulting in a loss of feature diversity (Papyan et al., 2020). Both phenomena are particularly relevant in the context of transfer learning, where models are initially trained on a source task before being fine-tuned for a target task. Our work, through the use of VCReg, seeks to mitigate these issues, offering a pathway to more effective transfer learning.

3 Variance-Covariance Regularization

3.1 Vanilla VCReg

Consider a labeled dataset comprising N samples, denoted as $\{(x_1, y_1) \dots (x_N, y_N)\}$ and a neural network $f_\theta(\cdot)$, which takes these inputs x_i and produces final predictions $\tilde{y}_i = f_\theta(x_i)$. In standard supervised learning, the loss is defined as $L_{\text{sup}} = \frac{1}{N}\sum_{i=1}^{N} \ell_{\text{sup}}(\tilde{y}_i, y_i)$.

The core objective of the Vanilla VCReg is to ensure that the D-dimensional input representations $\{h_i\}_{i=1}^N$ to the last layer of the network exhibit both high variance and low covariance. To achieve this, we employ variance and covariance regularization, same as mentioned in equation 1:

$$\ell_{\text{vcreg}}(h_1 \dots h_N) = \alpha \ell_{\text{var}}(h_1 \dots h_N) + \beta \ell_{\text{cov}}(h_1 \dots h_N) \tag{3}$$

Intuitively speaking, the covariance matrix captures the interdependencies among the dimensions of the feature vectors h_i. Maximizing ℓ_{var} encourages each feature dimension to contain unique, non-redundant information, while minimizing ℓ_{cov} aims to reduce the correlation between different dimensions, thus promoting feature independence. The overall training loss, which includes also the supervised loss, then becomes:

$$L_{\text{vanilla}} = \alpha \ell_{\text{var}}(h_1 \dots h_N) + \beta \ell_{\text{cov}}(h_1 \dots h_N) + \frac{1}{N}\sum_{i=1}^{N} \ell_{\text{sup}}(\tilde{y}_i, y_i). \tag{4}$$

Here, α and β serve as hyperparameters to control the strength of each regularization term.

3.2 Extending VCReg to Intermediate Representations

While regularizing the final layer in a neural network offers certain benefits, extending this approach to intermediate layers via VCReg provides additional advantages (for empirical evidence supporting this claim, please refer to Appendix A). Regularizing intermediate layers enables the model to capture more complex, higher-level abstractions. This strategy minimizes internal covariate shifts across layers, which in turn improves both the stability of training and the model's generalization capabilities. Furthermore, it fosters the development of feature hierarchies and enriches the latent space, leading to enhanced model interpretability and improved transfer learning performance.

To implement this extension, VCReg is applied at M strategically chosen layers throughout the neural network. For each intermediate layer j, we denote the feature representation for an input x_i as $h_i^{(j)} \in \mathbb{R}^{D_j}$. This culminates in a composite loss function, expressed as follows:

$$L_{\text{VCReg}} = \sum_{j=1}^{M} \left[\alpha \ell_{\text{var}}(h_1^{(j)} \dots h_N^{(j)}) + \beta \ell_{\text{cov}}(h_1^{(j)} \dots h_N^{(j)})\right] + \frac{1}{N}\sum_{i=1}^{N} \ell_{\text{sup}}(\tilde{y}_i, y_i). \tag{5}$$

Spatial Dimensions However, applying VCReg to intermediate layers of real-world neural networks presents challenges due to the spatial dimensions in these intermediate representations. Naively reshaping these representations into long vectors would lead to unmanageably large covariance matrices, thereby increasing computational costs and risking numerical instability. To address this issue, we adapt VCReg to accommodate networks with spatial dimensions. Each vector at a different spatial location is treated as an individual sample when calculating the covariance matrix. Both the variance loss and the covariance loss are then calculated based on this modified covariance matrix. In terms of practical implementation, VCReg is usually applied subsequently to each block within the neural network architecture, often succeeding residual connections. This placement allows for seamless incorporation into current network architectures and training paradigms.

Addressing Outliers with Smooth L1 Loss After treating spatial locations as independent samples for covariance computation, the resulting samples are no longer statistically independent. This can lead to outliers in the covariance matrix and unstable gradient updates. To address this, we introduce a smooth L1 penalty into the covariance loss term. Specifically, we replace the traditional squared covariance values C_{ij} in ℓ_{cov} with a smooth L1 function:

$$\text{SmoothL1}(x) = \begin{cases} x^2, & \text{if } |x| \le \delta \\ 2\delta|x| - \delta^2, & \text{otherwise} \end{cases} \tag{6}$$

Table 1: Transfer Learning Performance with ImageNet Supervised Pretraining
The table shows performance metrics for different architectures. Each model is pretrained on the full ImageNet dataset and then tested on different downstream datasets using linear probing. Application of VCReg consistently improves performance and beats other feature diversity regularizer. Averages are calculated excluding ImageNet results.

Architecture	iNat18	Places	Food	Cars	Aircraft	Pets	Flowers	DTD	Average
ResNet-50	42.8%	50.6%	69.1%	43.6%	54.8%	91.9%	77.1%	68.7%	62.33%
ResNet-50 (DeCov)	43.1%	50.4%	69.0%	45.7%	55.5%	90.6%	79.2%	69.1%	62.83%
ResNet-50 (WLD-Reg)	43.9%	**51.2%**	70.2%	43.9%	58.7%	91.4%	80.7%	69.0%	63.63%
ResNet-50 (VCReg)	**45.3%**	**51.2%**	**71.7%**	**54.1%**	**70.5%**	**92.1%**	**88.0%**	**70.8%**	**67.96%**
ConvNeXt-T	51.6%	53.8%	78.4%	62.9%	74.7%	93.9%	91.3%	72.9%	72.44%
ConvNeXt-T (VCReg)	**52.3%**	**54.7%**	**79.6%**	**64.2%**	**76.3%**	**94.1%**	**92.7%**	**73.3%**	**73.40%**
ViT-Base-32	39.1%	47.9%	70.6%	51.2%	63.8%	90.3%	84.6%	66.1%	64.20%
ViT-Base-32 (VCReg)	**40.6%**	**48.1%**	**70.9%**	**52.0%**	**65.8%**	**91.0%**	**86.6%**	**66.5%**	**65.19%**

By implementing this modification, we ensure that the loss function increases in a more controlled manner with respect to large covariance values. Empirically, this minimizes the impact of outliers, thereby enhancing the stability of the training process.

3.3 Fast Implementation

To optimize VCReg speed, we use the fact that VCReg only affects the loss function and not the forward pass. This allows us to focus on modifying the backward function for improvements. Specifically, we sidestep the usual process of calculating the VCReg loss and subsequent backpropagation. Instead, we directly adjust the computed gradients, which is feasible since the VCReg loss calculation relies solely on the current representation. Further details of this speed-optimized technique are outlined in Appendix B. Our optimized VCReg implementation exhibits similar latency as batch normalization layers and is more than 5 times faster than the naive VCReg implementation. The results are presented in Table 8.

4 Experiments

In this section, we first outline the experimental framework and findings highlighting the effectiveness of our proposed regularization approach, VCReg, within the realm of transfer learning that utilizes supervised pretraining for both images and videos. Subsequently, we extend our experiments to three specialized learning scenarios: 1) class imbalance via long-tail learning, 2) synergizing with self-supervised learning frameworks, and 3) hierarchical classification problems. The objective is to assess the adaptability of VCReg across various data distributions and learning paradigms, thereby evaluating its broader utility in machine learning applications. For details on reproducing our experiments, please consult Appendix C.

4.1 Transfer Learning for Images

In this section, we adhere to evaluation protocols established by seminal works such as (Chen et al., 2020; Kornblith et al., 2021; Misra & Maaten, 2020) for our transfer learning experiments. Initially, we pretrain models using three different architectures: ResNet-50 (He et al., 2016), ConvNeXt-Tiny (Liu et al., 2022), and ViT-Base-32 (Dosovitskiy et al., 2020), on the full ImageNet dataset. We follow the standard PyTorch recipes (Paszke et al., 2019) for all networks and do not modify any hyperparameters other than those related to VCReg to ensure a fair baseline comparison. Subsequently, we perform a linear probing evaluation across 9 different benchmark to evaluate the transfer learning performance. For ResNet-50, we include two other feature diversity regularizer methods for comparison: DeCov (Cogswell et al., 2015) and WLD-Reg (Laakom et al., 2023). We conduct experiments solely with ResNet-50 because it is the principal architecture used in the WLD-Reg paper. To ensure a fair comparison, we source hyperparameters from Laakom et al. (2023) for both DeCov and WLD-Reg.

Table 2: Transfer Learning Performance with Kinetics-400 and Kinetics-710 pretrained models: The table shows fine-tuning performance of Kinetics pre-trained models on HMDB51. VideoMAE-S, VideoMAE-B, and ViViT-B are pretrained on Kinetics-400 dataset while VideoMAEv2-S and VideoMAEv2-B are pre-trained on Kinetics-710. We apply VCReg only to the networks' output preceding the classification head. The results show that VCReg can boost the transfer learning classification performance for networks pre-trained on video data.

Method	Backbone	HMDB51
VideoMAE-S	ViT-S	79.9%
VideoMAE-S (VCReg)	ViT-S	**80.6%**
VideoMAE-B	ViT-B	82.2%
VideoMAE-B (VCReg)	ViT-B	**83.0%**
VideoMAEv2-S	ViT-S	83.6%
VideoMAEv2-S (VCReg)	ViT-S	**83.9%**
VideoMAEv2-B	ViT-B	86.5%
VideoMAEv2-B (VCReg)	ViT-B	**86.9%**
ViViT-B	ViT-B	70.9%
ViViT-B (VCReg)	ViT-B	**71.6%**

The results in Table 1 demonstrate that VCReg significantly enhances performance in transfer learning for images, achieving the highest performance for 9 out of 10 datasets, and for all three architectures. Clearly, VCReg acts as a versatile plug-in, effectively boosting transfer learning outcomes. Its effectiveness spans ConvNet and Transformer architectures, confirming its wide-ranging applicability.

4.2 Transfer Learning for Videos

To extend our evaluation of VCReg's efficacy, we conduct experiments using networks pretrained on video datasets. Specifically, we utilize models pretrained on Kinetics-400 Kay et al. (2017) and Kinetics-710 Li et al. (2022), subsequently finetuning them for action recognition on HMDB51 Kuehne et al. (2011). We experiment with models pretrained with self-supervised learning objectives (VideoMAE Tong et al. (2022) and VideoMAEv2 Wang et al. (2023)), as well as models pretrained with conventional supervised learning objectives (ViViT Arnab et al. (2021)). We follow the finetuning protocols detailed by Tong et al. (2022) and the conventional evaluation method used in the field, where the final performance is measured by the mean classification accuracy across three provided splits Simonyan & Zisserman (2014). To pinpoint the optimal VCReg coefficients, we conduct a grid search based on validation set accuracy. For simplicity, in this setup, VCReg regularization is exclusively applied to the final output of each network during finetuning, just before the classification head.

Table 2 illustrates that incorporating VCReg as a plugin regularizer improves the transfer learning performance for action recognition across various methods (VideoMAE, VideoMAE2, and ViViT-B) and backbone architectures (ViT-B and ViT-S). This solidifies VCReg's status as a practical and versatile regularizer, capable of substantially improving the performance of pretrained networks in transfer learning scenarios.

4.3 Class Imbalance with Long-Tail Learning

Class imbalance is a pervasive issue in many real-world datasets and poses a considerable challenge to standard neural network training algorithms. We conduct experiments to assess how well VCReg addresses this issue through long-tail learning. We evaluate VCReg using the CIFAR10-LT and CIFAR100-LT Krizhevsky et al. (2009) datasets, both engineered to have an imbalance ratio of 100. These experiments use a ResNet-32 backbone architecture. The per-class sample sizes ranges from 5,000 to 50 for CIFAR10-LT and from 500 to 5 for CIFAR100-LT.

Table 3 shows that models augmented with VCReg consistently outperform the standard ResNet-32 models on imbalanced datasets. These results are noteworthy because they demonstrate that VCReg effectively enhances the model's ability to discriminate between classes in imbalanced settings. This

Table 3: Performance Comparison on Class-Imbalanced Datasets Using VCReg: This table shows the accuracy of standard ResNet-32 with and without VCReg when trained on class-imbalanced CIFAR10-LT and CIFAR100-LT datasets. The VCReg-enhanced models show improved performance, demonstrating the method's effectiveness in addressing class imbalance.

Training Methods	CIFAR10-LT	CIFAR100-LT
ResNet-32	69.6%	37.4%
ResNet-32 (VCReg)	**71.2%**	**40.4%**

Table 4: Impact of VCReg on Self-Supervised Learning Methods: This table presents a comparative analysis of ResNet-50 models pretrained with SimCLR and VICReg losses on ImageNet, both with and without the VCReg applied. The models are evaluated using linear probing on various downstream task datasets. The VCReg models consistently outperform the non-VCReg models, showcasing the method's broad utility in transfer learning for self-supervised learning scenarios. Averages are calculated excluding ImageNet results.

Pretraining	ImageNet	iNat18	Places	Food	Cars	Aircraft	Pets	Flowers	DTD	Average
SimCLR	**67.2%**	37.2%	52.1%	66.4%	35.7%	**62.3%**	76.3%	82.6%	68.1%	60.09%
SimCLR+VCR	67.1%	**41.3%**	**52.3%**	**67.7%**	**40.6%**	61.9%	**76.6%**	**83.6%**	**69.0%**	**61.63%**
VICReg	65.2%	**41.7%**	48.2%	61.0%	27.3%	51.2%	79.1%	74.3%	65.4%	56.03%
VICReg+VCR	**66.3%**	41.4%	**49.6%**	**61.6%**	**29.3%**	**54.2%**	**79.7%**	**74.5%**	**66.5%**	**57.10%**

establishes VCReg as a valuable tool for real-world applications where class imbalance is often a concern.

4.4 Self-Supervised Learning with VCReg

Our subsequent investigation focuses on examining the synergy between VCReg and existing self-supervised learning paradigms. As mentioned in the previous sections, we apply VCReg not only to the final but also to intermediate representations. So in all of the following experiments for self-supervised learning with VCReg, we apply the original loss function to the output of the network, and the VCReg loss to all the intermediate representations. We employ a ResNet-50 architecture, training it for 100 epochs under four different configurations: using either SimCLR loss or VICReg loss, coupled with the ImageNet dataset. For evaluation, we conduct linear probing tests on multiple downstream task datasets, following the protocols prescribed by Misra & Maaten (2020); Zbontar et al. (2021).

As indicated in Table 4, integrating VCReg into self-supervised learning paradigms such as SimCLR and VICReg results in consistent performance improvements for transfer learning. Specifically, the linear probing accuracies are enhanced across nearly all the evaluated datasets. These gains underscore the broad applicability and versatility of VCReg, demonstrating its potential to enhance various machine learning methodologies.

4.5 Hierarchical Classification

To evaluate the efficacy of the learned representations across multiple levels of class granularity, we conduct experiments on the CIFAR100 dataset as well as five distinct subsets of ImageNet (Engstrom et al., 2019). In each dataset, every data sample is tagged with both superclass and subclass labels, denoted as $(x_i, y_i^{\text{sup}}, y_i^{\text{sub}})$. Note that while samples sharing the same subclass label also share the same superclass label, the reverse does not necessarily hold true. Initially, the model is trained using only the superclass labels, i.e., the (x_i, y_i^{sup}) pairs. Subsequently, linear probing is employed with the subclass labels (x_i, y_i^{sub}) to assess the quality of abstract features at the superclass level.

Table 5 presents key performance metrics, highlighting the substantial improvements VCReg brings to subclass classification. The improvements are consistent across all datasets, with the CIFAR100 dataset showing the most significant gain—an increase in accuracy from 60.7% to 72.9%. These results underscore VCReg's capability to assist neural networks in generating feature representations that are not only discriminative at the superclass level but are also well-suited for subclass distinctions.

Table 5: Impact of VCReg on Hierarchical Classification in ConvNeXt Models: This table summarizes the classification accuracies obtained with ConvNeXt models, both with and without the VCReg regularization, across multiple datasets featuring hierarchical class structures including CIFAR100 and several subsets of ImageNet. The models were initially trained using superclass labels and subsequently probed using subclass labels. VCReg consistently boosts performance in subclass classification tasks.

	CIFAR100	living_9	mixed_10	mixed_13	geirhos_16	big_12
Superclass Count	20	9	10	13	16	12
Subclass Count	100	72	60	78	32	240
ConvNeXt	60.7%	53.4%	60.3%	61.1%	60.5%	51.8%
ConvNeXt (VCReg)	**72.9%**	**62.2%**	**67.7%**	**66.0%**	**70.1%**	**61.5%**

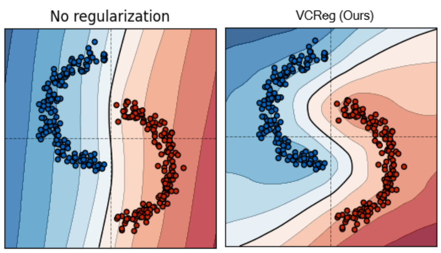


Figure 2: Comparative evaluation between training with and without VCReg on a "Two-Moon" Synthetic Dataset. Decision boundaries are averaged over ten distinct runs with random data point sampling and model initialization. A single run's data points are displayed for clarity. While "No regularization" has limitations in forming intricate decision boundaries, VCReg is effective in generating meaningful ones.

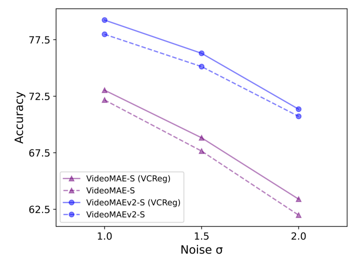


Figure 3: Impact of VCReg amidst noisy data: This figure shows the top-1 accuracy of VideoMAE-S and VideoMAEv2-S when fine-tuned for action recognition using HMDB51 corrupted with synthetic noise. We corrupt the data with Gaussian noise with standard deviation $\sigma \in \{1, 1.5, 2\}$. Models with VCReg outperform their non-regularized counterparts in this setting.

This attribute is particularly advantageous in real-world applications where class categorizations often exist within a hierarchical framework.

5 Exploring the Benefits of VCReg

This section aims to thoroughly unpack the multi-faceted benefits of VCReg in the context of supervised neural network training. Specifically, we discuss its capability to address challenges such as gradient starvation (Pezeshki et al., 2021), neural collapse (Papyan et al., 2020), noisy data, and the preservation of information richness during model training (Shwartz-Ziv, 2022).

5.1 Mitigating Gradient Starvation

In line with the original study on gradient starvation (Pezeshki et al., 2021), we observe that most traditional regularization techniques fall short of capturing the vital features for the "two-moon" dataset experiment. To assess the effectiveness of VCReg, we replicate this setting with a three-layer network and apply our method during training. Our visualized results in Figure 2 make it apparent that VCReg has a marked advantage over traditional regularization techniques, particularly in the aspects of separation margins. Thus, it is reasonable to conclude that VCReg can help mitigate gradient starvation. Please check section E for the detailed information about experiments related to the "two-moon" dataset.

5.2 Preventing Neural Collapse and Information Compression

To deepen our understanding of VCReg and its training dynamics, we closely examine its learned representations. A recent study (Papyan et al., 2020) observed a peculiar trend in deep networks trained for classification tasks: the top-layer feature embeddings of training samples from the same

Table 6: VCReg learns richer representation and prevents neural collapse and information compression Metrics include Class-Distance Normalized Variance (CDNV), Nearest Class-Center Classifier (NCC), and Mutual Information (MI). Higher values indicate reduced neural collapse and richer feature representations.

Network	CDNV	NCC	MI
ConvNeXt	0.28	0.99	2.8
ConvNeXt (VCReg)	**0.56**	**0.81**	**4.6**

class tend to cluster around their respective class means, which are as distant from each other as possible. However, this phenomenon could potentially result in a loss of diversity among the learned features (Papyan et al., 2020), thus curtailing the network's capacity to grasp the complexity of the data and leading to suboptimal performance for transfer learning (Li et al., 2018). Our neural collapse investigation includes two key metrics.

Class-Distance Normalized Variance (CDNV) For a feature map $f : \mathbb{R}^d \to \mathbb{R}^p$ and two unlabeled sets of samples $S_1, S_2 \subset \mathbb{R}^d$, the CDNV is defined as

$$V_f(S_1, S_2) = \frac{\sigma_f^2(S_1) + \sigma_f^2(S_2)}{2\|\mu_f(S_1) - \mu_f(S_2)\|^2}, \tag{7}$$

where $\mu_f(S)$ and $\sigma_f^2(S)$ signify the mean and variance of the set $\{f(x) \mid x \in S\}$. This metric measures the degree of clustering of the features, in relation to their distance.

Nearest Class-Center Classifier (NCC) This classifier is defined as $\arg\min_{c\in[C]} \|f(x) - \mu_f(S_c)\|$

According to this measure, during training, collapsed feature embeddings in the penultimate layer become separable, and the classifier converges to the "nearest class-center classifier".

Preventing Information Compression Although effective compression often yields superior representations, overly aggressive compression might cause the loss of crucial information about the target task (Shwartz-Ziv et al., 2018; Shwartz-Ziv & Alemi, 2020; Shwartz-Ziv & LeCun, 2023). To investigate the compression during the learning, we use the mutual information neural estimation (MINE) (Belghazi et al., 2018), a method specifically designed to estimate the mutual information between the input and its corresponding embedded representation. This metric effectively gauges the complexity level of the representation, essentially indicating how much information it encodes.

We evaluate the learned representations of two ConvNeXt models (Liu et al., 2022), which are trained on ImageNet with supervised loss. One model is trained with VCReg, while the other is trained without. As demonstrated in Table 6, both types of collapse, measured by CDNV and NCC, and the mutual information reveal that VCReg representations have significantly more diverse features and information compared to regular training. This suggests that the VCReg mitigates the neural collapse and prevents excessive information compression, two crucial factors that often limit the effectiveness of deep learning models in transfer learning tasks. Our findings highlight the potential of VCReg as a valuable addition to the deep learning toolbox, significantly increasing the generalizability of learned representations.

5.3 Providing Robustness to Noise

In real-world scenarios, encountering noise is a common challenge, making robustness against noise a crucial feature for any effective transfer learning algorithm. Recognizing the ubiquity of noise in practical applications, we aim to evaluate the capability of VCReg to bolster transfer learning performance in noisy environments. For this purpose, we utilize video networks initially pretrained on Kinetics-400 and Kinetics-710, as mentioned in section 4.2. We then finetune these networks on the HMDB51 dataset, which is deliberately subjected to varying levels of Gaussian noise. The findings in Figure 3 reveal a clear advantage: incorporating VCReg notably improves the resilience of VideoMAE-S and VideoMAEv2-S models to noisy data. Appendix D shows that this trend of increased durability against noise is consistently seen in larger models, such as VideoMAE-B and VideoMAEv2-B.

6 Conclusion

In this work, we address prevalent challenges in supervised pretraining for transfer learning by introducing an efficient and adaptable regularization technique called Variance-Covariance Regularization (VCReg). Our comprehensive evaluation revels that using VCReg yields significant improvements in transfer learning performance across various network architectures, learning paradigms, and data modalities. Moreover, our in-depth analysis confirms VCReg's effectiveness in overcoming typical transfer learning hurdles such as neural collapse, gradient starvation, and noisy data. Our work paves the way for further research to achieve highly optimized and generalizable machine learning models.

References

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6836–6846, 2021.

Babajide O Ayinde, Tamer Inanc, and Jacek M Zurada. Regularizing deep neural networks by enhancing diversity in feature extraction. *IEEE transactions on neural networks and learning systems*, 30(9):2650–2661, 2019.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

Ido Ben-Shaul, Ravid Shwartz-Ziv, Tomer Galanti, Shai Dekel, and Yann LeCun. Reverse engineering self-supervised learning. *arXiv preprint arXiv:2305.15614*, 2023.

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36. JMLR Workshop and Conference Proceedings, 2012.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pp. 446–461. Springer, 2014.

Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

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

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. URL `https://github.com/MadryLab/robustness`.

Jonas Geiping, Micah Goldblum, Gowthami Somepalli, Ravid Shwartz-Ziv, Tom Goldstein, and Andrew Gordon Wilson. How much data are augmentations worth? an investigation into scaling laws, invariance, and implicit regularization. *arXiv preprint arXiv:2210.06441*, 2022.

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.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.

Simon Kornblith, Ting Chen, Honglak Lee, and Mohammad Norouzi. Why do better loss functions lead to less transferable features? *Advances in Neural Information Processing Systems*, 34: 28648–28662, 2021.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pp. 2556–2563. IEEE, 2011.

Firas Laakom, Jenni Raitoharju, Alexandros Iosifidis, and Moncef Gabbouj. Wld-reg: A data-dependent within-layer diversity regularizer. *arXiv preprint arXiv:2301.01352*, 2023.

Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–50. Springer, 2002.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer, 2022.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6707–6717, 2020.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, 2008.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272, 2021.

Ravid Shwartz-Ziv. Information flow in deep neural networks. *arXiv preprint arXiv:2202.06749*, 2022.

Ravid Shwartz-Ziv and Alexander A Alemi. Information in infinite ensembles of infinitely-wide neural networks. In *Symposium on Advances in Approximate Bayesian Inference*, pp. 1–17. PMLR, 2020.

Ravid Shwartz-Ziv and Yann LeCun. To compress or not to compress–self-supervised learning and information theory: A review. *arXiv preprint arXiv:2304.09355*, 2023.

Ravid Shwartz-Ziv, Amichai Painsky, and Naftali Tishby. Representation compression and generalization in deep neural networks, 2018.

Ravid Shwartz-Ziv, Randall Balestriero, and Yann LeCun. What do we maximize in self-supervised learning? *arXiv preprint arXiv:2207.10081*, 2022.

Ravid Shwartz-Ziv, Randall Balestriero, Kenji Kawaguchi, Tim GJ Rudner, and Yann LeCun. An information-theoretic perspective on variance-invariance-covariance regularization. *arXiv preprint arXiv:2303.00633*, 2023.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14549–14560, 2023.

Karl R. Weiss, Taghi M. Khoshgoftaar, and Dingding Wang. A survey of transfer learning. *Journal of Big Data*, 3, 2016.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

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.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. corr abs/1611.03530 (2016). *arXiv preprint arxiv:1611.03530*, 2016.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *Advances in neural information processing systems*, 27, 2014.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

A EXPERIMENTAL INVESTIGATION ON EFFECTIVE APPLICATION OF VCREG TO STANDARD NETWORKS

To determine the optimal manner of integrating the VCReg into a standard network, we conducted several experiments utilizing the ConvNeXt-Atto architecture, trained on ImageNet following the torchvision (Paszke et al., 2019) training recipe. To reduce the training time, we limited the network training to 90 epochs with a batch size of 4096. The complete configuration comprised 90 epochs, a batch size of 4096, two learning rate of $\{0.016, 0.008\}$ with a 5 epochs linear warmup followed by a cosine annealing decay. The weight decay was set at 0.05 and the norm layers were excluded from the weight decay. we experimented with $\alpha \in \{1.28, 0.64, 0.32, 0.16\}$ and $\beta \in \{0.16, 0.08, 0.04, 0.02, 0.01\}$.

We experimented with incorporating the VCReg layers in four different locations:

1. Applying the VCReg exclusively to the second last representation (the input of the classification layer).
2. Applying VCReg to the output of each ConvNeXt block.
3. Applying VCReg to the output of each downsample layer.
4. Applying VCReg to the output of both, each ConvNeXt block and each downsample layer.

The VCReg layer was implemented as detailed in 1, with the addition of a mean removal layer along the batch preceding the VCReg layer to ensure that the VCReg input exhibited a zero mean.

Table 7: Transfer Learning Experiments with Different VCReg Configurations

Architecture	Food	Cars	Aircraft	Pets	Flowers	DTD
ConvNeXt-Atto (VCReg1)	63.2%	39.6%	55.9%	89.1%	85.3%	65.1%
ConvNeXt-Atto (VCReg2)	**66.8**%	48.1%	**60.4**%	**91.1**%	**86.4**%	**66.4**%
ConvNeXt-Atto (VCReg3)	64.0%	40.9%	56.5%	89.4%	85.9%	65.1%
ConvNeXt-Atto (VCReg4)	66.7%	**48.3**%	59.6%	90.6%	85.6%	66.1%

The results in Table 7 indicate superior performance when the VCReg layer is applied to the output of each block (second setup) or applied to the output of blocks and downsample layers (fourth setup) compared to the other setups. Considering architectures like ViT lack downsample layers, for consistency across different architectures, we decided to use the second configuration for further experiments.

B THE FAST IMPLEMENTATION OF THE VCREG

The VCRegeg does not affect the forward pass in any way, allowing us to substantially speed up the implementation by modifying the backward function directly. Instead of computing the VCReg loss and backpropagating it, we can directly alter the calculated gradient. This is possible since the VCReg loss calculation only requires the current representation. The specifics of this speed-optimized implementation are outlined in Algorithm 1.

We quantify the computational overhead by measuring the average time required for one NVIDIA A100 GPU to execute both the forward and backward passes on the entire network for a batch size of 128 using the ImageNet dataset. These results are summarized in Table 8. For the sake of comparison, we also include the latencies associated with adding Batch Normalization (BN) layers, revealing that our optimized VCReg implementation exhibits similar latencies to BN layers and is almost 5 times faster than the naive implementation.

Algorithm 1: PyTorch-Style Pseudocode for Fast VCReg Implementation

```
# α, β and ε :  hyperparameters
# mm:  matrix-matrix multiplication

class VarianceCovarianceRegularizationFunction(Function):
    # forward pass
    # We assume the input has zero mean per channel
    # In practice, we apply a batch demean operation before calling the function
    def forward(ctx, input):
        ctx.save_for_backward(input)
        return input
    # backward pass
    def backward(ctx, grad_output):
        input, = ctx.saved_tensors
        # reshape the input to have (n, d) shape
        flattened_input = input.flatten(start_dim=0, end_dim=-2)
        n, d = flattened_input.shape
        # calculate the covariance matrix
        covariance_matrix = mm(flattened_input.t(), flattened_input) / (n - 1)
        # calculate the gradient
        diagonal = F.threshold(rsqrt(covariance_matrix.diagonal() + \epsilon), 1.0, 0.0)
        std_grad_input = diagonal * flattened_input
        cov_grad_input = torch.mm(flattened_input, covariance_matrix.fill_diagonal_(0))

        grad_input = grad_output
                  - α/(d(n − 1)) * std_grad_input.view(grad_output)
                  + 4β/(d(d − 1)) * cov_grad_input

        return grad_input
```

Table 8: Average Time Required for One Forward and Backward Pass with Various Layers Inserted Comparison of computational latencies across different configurations of ViT and ConvNeXt networks. The table demonstrates the efficacy of the optimized VCReg layer in terms of computational time, compared to both naive VCReg and Batch Normalization (BN) layers.

Network	Number of Inserted Layers	Identity	VCReg (Naive)	VCReg (Fast)	BN
ViT-Base-32	12	0.223s	1.427s	0.245s	0.247s
ConvNeXt-T	18	0.442s	2.951s	0.471s	0.468s

C Implementation Details

C.1 Transfer Learning Experiments with ImageNet Pretraining

In conducting the transfer learning experiments, we adhered primarily to the training recipe specified by PyTorch Paszke et al. (2019) for each respective architecture during the supervised pretraining phase. We abstained from pretraining any of the baseline models, instead opting to directly download the weights from PyTorch's own repository. The only modifications applied were to the parameters associated with VCReg loss, and we experimented with $\alpha \in \{1.28, 0.64, 0.32, 0.16\}$ and $\beta \in \{0.16, 0.08, 0.04, 0.02, 0.01\}$.

For iNaturalist 18 Van Horn et al. (2018) and Place205 Zhou et al. (2014), we relied on the experimental settings detailed in Zbontar et al. (2021) for the linear probe evaluation.

Regarding Food-101 Bossard et al. (2014), Stanford Cars Krause et al. (2013), FGVC Aircraft Maji et al. (2013), Oxford-IIIT Pets Parkhi et al. (2012), Oxford 102 Flowers Nilsback & Zisserman (2008), and the Describable Textures Dataset (DTD) Cimpoi et al. (2014), we complied with the evaluation protocol provided by Chen et al. (2020); Kornblith et al. (2021). An $L2$-regularized multinomial logistic regression classifier was trained on features extracted from the frozen pretrained network. Optimization of the softmax cross-entropy objective was conducted using L-BFGS, without the application of data augmentation. All images were resized to 224 pixels along the shorter side through bicubic resampling, followed by a 224 x 224 center crop. The $L2$-regularization parameter was selected from a range of 45 logarithmically spaced values between 0.00001 and 100000.

All experiments were run three times, with the average results presented in Table 1.

C.2 Transfer Learning Experiments with Kinetics pre-trained Models

In conducting experiments with video-pretrained models, we utilize the publicly available code bases and model checkpoints provided for VideoMAE and VideoMAEv2 (`https://github.com/MCG-NJU/VideoMAE` and `https://github.com/OpenGVLab/VideoMAEv2`). For both VideoMAE and VideoMAEv2 we use ViT-Small and ViT-Base checkpoints. VideoMAE models are pre-trained on Kinetics-400 while VideoMAEv2 on Kinetics-710. We use the pre-trained checkpoint for ViViT-B (ViT-Base backbone) pre-trained on Kinetics-400 from HuggingFace. For evaluation, we adopt the inference protocol of 10 clips $\times$ 3 crops. For VCReg hyperparameters experiments with values for $\alpha \in 1, 3, 5$ and $\beta \in \{0.1, 0..3, 0.5\}$. For the rest of the finetuning hyperparameters as well as the data pre-processing and evaluation protocol, we use the configuration for HMDB51 available in VideoMAE Tong et al. (2022) and its corresponding code base (linked above).

C.3 Subclass Linear Probing Result with Network Pretrained on Superclass Label

For our subclass linear probing experiments, we employed a ConvNeXt-Atto network. Each model was pretrained for 200 epochs using the superclasses, adhering to the same procedure detailed in the Appendix A. Subsequent to this pretraining phase, we initiated a linear probing process using the subclass labels. This linear classifier was trained for 100 epochs, using a base learning rate of 0.016 in conjunction with a cosine learning rate schedule. The optimizer used was AdamW, which worked to minimize cross-entropy loss with a weight decay set at 0.05. We processed our training data in batches of 256.

C.4 Long-Tail Learning Result

For our long-tail learning experiments, we use ResNet-32 as a backbone for experiments on the CIFAR10-LT and CIFAR100-LT datasets. We trained 100 epochs with batch size 256, Adam optimizer with two learning rate of $\{0.016, 0.008\}$ with a 10-epoch linear warm-up followed by a cosine annealing decay. The weight decay was set at 0.05 and the norm layers were excluded from the weight decay. we experimented with $\alpha \in \{1.28, 0.64, 0.32, 0.16\}$ and $\beta \in \{0.16, 0.08, 0.04, 0.02, 0.01\}$.

C.5 VCReg with Self-Supervised Learning Methods

We train a ResNet-50 model in four different setups, using either the SimCLR loss or the VICReg loss with the ImageNet dataset. The application of the VCReg is the same as described in Appendix A.

We closely follow the original setting in Chen et al. (2020) for SimCLR pretraining and Bardes et al. (2021) for VICReg pretraining.

Augmentation For both methods, we use the same augmentation methods. Each augmented view is generated from a random set of augmentations of the same input image. We apply a series of standard augmentations for each view, including random cropping, resizing to 224x224, random horizontal flipping, random color-jittering, randomly converting to grayscale, and a random Gaussian blur. These augmentations are applied symmetrically on two branches Geiping et al. (2022)

Architecture For SimCLR, the encoder is a ResNet-50 network without the final classification layer followed by a projector. The projector is a two-layer MLP with input dimension 2048, hidden dimension 2048, and output dimension 256. The projector has ReLU between the two layers and batch normalization after every layer. This 256-dimensional embedding is fed to the infoNCE loss.

For VICReg, the online encoder is a ResNet-50 network without the final classification layer. The online projector is a two-layer MLP with input dimension 2048, hidden dimension 8192, and output dimension 8192. The projector has ReLU between the two layers and batch normalization after every layer. This 8192-dimensional embedding is fed to the infoNCE loss.

For VCReg, we just applied the VCReg layers to the ResNet-50 network as described in the Appendix A.

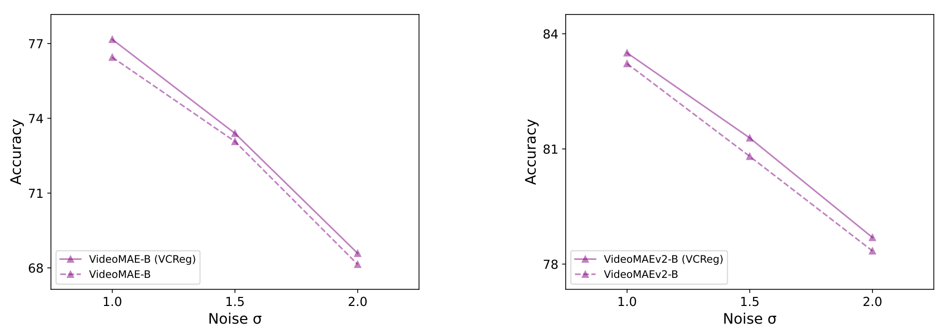
Optimization We follow the training protocol in Zbontar et al. (2021). For SimCLR experiments, we used a LARS optimizer and a base learning rate 0.3 with cosine learning rate decay schedule. We pretrain the model for 100 epochs with 5 epochs warm-up with batch size 4096.

For VICReg, we use a LARS optimizer and a base learning rate 0.2 using cosine learning rate decay schedule. We pretrain the model for 100 epochs with 5 epochs warm-up with batch size 4096.

Evaluation We follow the standard evaluation protocol as prescribed by Misra & Maaten (2020); Zbontar et al. (2021), performing linear probing evaluations, on iNaturalist 18 Van Horn et al. (2018) and Place205 Zhou et al. (2014) datasets.

D Robustness to noise

This section provides additional results on measuring VCReg's ability to enhance transfer learning performance in the presence of noise. In these experiments we start with VideoMAE-B and VideoMAEv2-B networks (from section 4.2) pre-trained on Kinetics-400 and Kinetics-710, respectively, then fine-tune them on HMDB51 corrupted with varying levels of Gaussian noise. During fine-tuning, we compare the transfer learning performance of VideoMAE-B and VideoMAEv2-B networks with and without the addition of VCReg. When VCReg is added, it is only applied to the final layer of these networks preceding the classification head. Figure 4 shows that VCReg models outperform their non-regularized counterparts in this setting.



(a) VideoMAE-B (b) VideoMAEv2-B

Figure 4: Impact of VCReg amidst noisy data: This figure shows the top-1 accuracy of VideoMAE-B and VideoMAEv2-B when fine-tuned for action recognition using HMDB51 with synthetic noise. We corrupt the data with Gaussian noise with standard deviation $\sigma \in \{1, 1.5, 2\}$. Models with VCReg outperform their non-regularized counterparts in this setting.

E Two-Moon Dataset

In alignment with the original gradient starvation study Pezeshki et al. (2021), we notice that most regular routine regularization techniques do not sufficiently capture the necessary features for the "two-moon" dataset experiment. To evaluate our approach, we mirrored this setting and applied the VCReg during the training.

The synthetic "two-moon" dataset comprises two classes of points, each forming a moon-like shape. The gradient starvation study highlighted an issue where if the gap between the two moons is wide enough for a straight line to separate the two classes, the network stops learning additional features and focuses solely on a single feature. We duplicated this situation using a three-layer network and applied all the initially tested methods in the original study. The resulting decision boundary after training with the "two-moon" dataset is visualized in Figure 5.

From the visualization, it becomes apparent that not only does VCReg outperform other conventional regularization techniques in separation margins, but also it shows superior performance compared to spectral decoupling, a method specifically designed for this task. VCReg is effective in maximizing

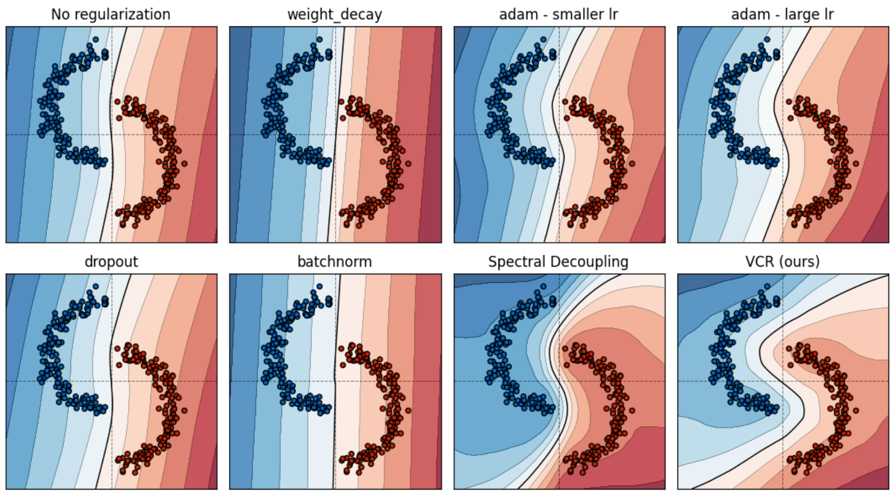


Figure 5: The effect of conventional regularization methods and the VCReg on a simple task of two-moon classification. Shown decision boundaries are the average over 10 runs in which data points and the model initialization parameters are sampled randomly. Here, only the data points of one particular seed are plotted for visual clarity. It can be seen that conventional regularizations of deep learning seem not to help with learning a curved decision boundary.

the variance while minimizing the covariance in the feature space, an achievement that is not obtained by other techniques such as L2, dropout Hinton et al. (2012), and batch normalization Ioffe & Szegedy (2015). Consequently, these other techniques yield features that are less discriminative and informative.

F Compute Resources

The majority of our experiments were run using AMD MI50 GPUs. The longest pretraining for ConvNeXt-Tiny takes about 48 hours on 2 nodes, where each node has 8 MI50 GPUs attached. We estimate that the total amount of compute resources used for all the experiments can be roughly approximated by 60 (days) $\times$ 24 (hours per day) $\times$ 8 (nodes) $\times$ 8 (GPUs per nodes) $= 92,160$ (GPU hours).

We are aware of potential environmental impact of consuming a lot of compute resources needed for this work, such as atmospheric CO_2 emissions due to the electricity used by the servers. However, we also believe that advancements in representation learning and transfer learning can potentially help mitigate these effects by reducing the need for data and compute resources in the future.

G Limitations

Due to a lack of compute resources, we were unable to conduct a large number of experiments with the goal of tuning hyperparameters and searching for the best configurations. Therefore, the majority of hyperparameters and network configurations used in this work are the same as provided by PyTorch Paszke et al. (2019). The only hyperparameters that were tuned were α and β, the coefficients for VCR. All the other hyperparameters may not be optimal.

In addition, all models were pretrained on the ImageNet Deng et al. (2009) and Krizhevsky et al. (2009) dataset, so their performances might differ if pretrained with other datasets containing different data distributions or different types of images (e.g., x-rays). We encourage further exploration in this direction for current and future self-supervised learning frameworks.