CCA with Shared Weights for Self-Supervised Learning

James Chapman[*]
Department of Computer Science
University College London
90 High Holborn, London
james.chapman.19@ucl.ac.uk

Lennie Wells[*]
Faculty of Mathematics
University of Cambridge
Wilberforce Road
ww347@cam.ac.uk

Abstract

In this paper, we explore SSL-EY (**S**elf-**S**upervised **L**earning with an **E**ckhart-**Y**oung characterisation), a novel self-supervised learning loss function directly inspired by Deep Canonical Correlation Analysis (DCCA). Our key insight is that maximizing the correlation of learned representations can serve as an effective and interpretable objective in self-supervised learning. We demonstrate that SSL-EY not only strengthens the theoretical underpinning of existing methods, such as Barlow Twins and VICReg, but also performs competitively on benchmark datasets.

1 Introduction

Self-Supervised Learning (SSL) methods have reached the state of the art in tasks such as image classification [4]. These methods can learn robust data representations without the need for explicit labels or supervision. Recently, a family of SSL methods that are closely aligned with Canonical Correlation Analysis (CCA) has garnered interest. This family notably includes Barlow Twins [23], VICReg [6], and W-MSE [11] and they aim to transform a pair of data views into similar representations, similar to the objective of CCA. Similarly, some generative approaches to SSL[18] bear a striking resemblance to Probabilistic CCA[3]. These connections have started to be explored in [5].

Deep CCA [2] secured a runner-up position for the test-of-time award at ICML 2023 [13]. However, its direct application has been limited in large datasets due to biased gradients in the stochastic minibatch setting. There have since been proposals to scale-up Deep CCA in the stochastic case with adaptive whitening [22] and regularization [8], but these techniques are highly sensitive to hyperparameter tuning.

This work expands on recent work [9], which introduced a new SSL loss function based on Deep CCA. This loss function has no tuning parameters in its objective and is interpretable via sums of squared correlations. Moreover, the learnt representations are provably full rank in the linear setting, in contrast to VICReg and Barlow twins [9]; they therefore do not suffer from collapse in the deep setting[2]. In Section 4, we demonstrate the practicality of our method by applying our

[*] Equal contribution.

[2] Reasoning through a heuristic argument, with empirical support.

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

loss function to the standard CIFAR-10 and CIFAR-100 benchmarks. We match the state-of-the-art in terms of accuracy while using the default hyperparameters from a fully optimized Barlow Twins model. Additionally, we experiment with varying the projector size to demonstrate that, when directly optimizing for CCA, we can achieve the same performance with a much smaller embedding size and even eliminate the need for a projector altogether.

2 Background: A Unified Approach to CCA and SSL Family

Multi-view setting: The initial motivation for this work was from a multi-view learning setting where there are I different sets of observations, referred to as 'views'[3], corresponding to the same individuals. We seek to learn lower dimensional representations of the data that reflect the information shared between the different views and can be used for various downstream tasks, such as classification or similarity preservation. To formalise this setting, we model observations as realisations of a collection of vector-valued random variables $X^{(i)} \in \mathbb{R}^{D_i}$ for $i \in \{1, \dots, I\}$ and seek to learn K-dimensional representations of the form:

$$Z^{(i)} = f^{(i)}(X^{(i)}; \theta^{(i)}), \tag{1}$$

where each function $f^{(i)}$ is differentiable in its parameters $\theta^{(i)}$ for any given input $X^{(i)}$. For simplicity, we will only consider the two-view case $I = 2$ for the rest of this work, but we note that the approach immediately generalises to any number of views I, see [9].

An unbiased loss function for stochastic CCA: It is well known that CCA [7] can be defined by the solution to the generalized eigenvalue problem (GEP) $Au = \lambda Bu$ where:

$$A = \begin{pmatrix} 0 & \operatorname{Cov}(X^{(1)}, X^{(2)}) \\ \operatorname{Cov}(X^{(2)}, X^{(1)}) & 0 \end{pmatrix}, \quad B = \begin{pmatrix} \operatorname{Var}(X^{(1)}) & 0 \\ 0 & \operatorname{Var}(X^{(2)}) \end{pmatrix}, \quad u = \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix}. \tag{2}$$

By applying the Eckhart–Young inequality [20] to the eigen-decomposition of $B^{-1/2} A B^{-1/2}$, [9] showed that top-eigenspaces of GEPs can be characterised as minimising the 'Eckhart-Young' loss:

$$\mathcal{L}_{\text{EY}}(U) := \operatorname{trace}\left(-2\, U^T A U + \left(U^T B U\right)\left(U^T B U\right)\right) \tag{3}$$

where the columns of $U \in \mathbb{R}^{D \times K}$ span a K-dimensional subspace of $\mathbb{R}^D$ associated with the top-K eigenvalues. Applying this characterisation to eq. (2) in the linear case $Z^{(i)} = {U^{(i)}}^T X^{(i)}$ motivates the following loss for CCA that can immediately be generalised to the Deep (and Multiview) setting:

$$\mathcal{L}_{\text{EY}}(\theta) = -2 \operatorname{trace} C(\theta) + \|V(\theta)\|_F^2. \tag{4}$$

where $C(\theta) = \sum_{i \neq j} \operatorname{Cov}(Z^{(i)}, Z^{(j)})$, and $V(\theta) = \sum_i \operatorname{Var}(Z^{(i)})$ are sums of between-view and within-view variances respectively.

Siamese Networks and Shared Weights in SSL: Siamese networks are commonly used in unimodal SSL. In these setups, an original input datum $X^{(0)}$ is typically modified by a pair of independent random *augmentations* to obtain a pair of augmented views $(X^{(1)}, X^{(2)})$. These pass through identical encoders to form embeddings $(Z^{(1)}, Z^{(2)})$; i.e. we have $f^{(1)} = f^{(2)} = f$ and have shared weights $\theta^{(1)} = \theta^{(2)} = \theta$ in eq. (1).

Classical CCA does not assume shared weights between different views, but does recover shared weights if pairs of data are generated by independent random augmentations as above; the same is true of VICReg in the linear case [9]. Weight sharing can therefore be seen as helpful regularisation, which is why we suggest using it in this setting.

Joint Embedding for SSL and the Role of the Projector: Many recent SSL methods, including Barlow Twins and VICReg, use an encoder-projector setup, as illustrated in Figure 1. Input data is mapped through an *encoder* g to obtain *representations*; these representations are then mapped through a *projector*[4] h to form (typically) higher-dimensional *embeddings*. Crucially, it is the representations that are used for downstream tasks, while the embeddings are used to train the model.

[3] Yes, there are I (eye) views

[4] Sometimes alternatively called an *expander*.

Encoder-projector architectures have had impressive empirical success, but despite recent work [17, 14], there is relatively little understanding of why they perform so well.

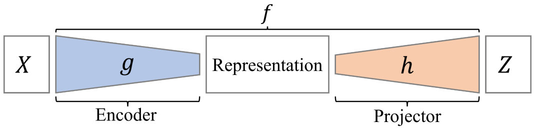


Figure 1: A schematic diagram of the architecture used by Joint Embedding methods which include VICReg, and Barlow Twins

3 Our Approach: Deep CCA with Shared Weights

Building on these insights, we propose a novel approach that directly applies Deep CCA (DCCA) with shared weights to SSL.

Unbiased estimates: since empirical covariance matrices are unbiased, we can construct unbiased estimates to C, V from a batch of transformed variables $\mathbf{Z} = (\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)})$[5] via

$$\hat{C}(\theta)[\mathbf{Z}] = \widehat{\mathrm{Cov}}(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}) + \widehat{\mathrm{Cov}}(\mathbf{Z}^{(2)}, \mathbf{Z}^{(1)}),\ \hat{V}(\theta)[\mathbf{Z}] = \widehat{\mathrm{Var}}(\mathbf{Z}^{(1)}) + \widehat{\mathrm{Var}}(\mathbf{Z}^{(2)}) \tag{5}$$

Then if $\mathbf{Z}, \mathbf{Z}'$ are two independent batches of transformed variables[6], the batch loss

$$\hat{\mathcal{L}}_{\mathrm{EY}}[\mathbf{Z}, \mathbf{Z}'] := -2 \operatorname{trace} \hat{C}[\mathbf{Z}] + \langle \hat{V}[\mathbf{Z}], \hat{V}[\mathbf{Z}'] \rangle_F \tag{6}$$

gives an unbiased estimate of $\mathcal{L}_{\mathrm{EY}}(\theta)$. Our approach has the following interesting properties:

- **Interpretable Loss:** The negative of the loss value is precisely the sum of squared canonical correlations between the pair of embeddings (under mild conditions [9]). This can be used as an interpretable metric for the quality of a representation and the saturation of the embedding space (i.e. if the correlations are all perfect then we have capacity to learn more information).
- **Provably full rank embeddings:** The embeddings are provably full rank in the linear case, unlike VICReg and Barlow twins [9]; they are therefore also full rank in general in the deep case, so do not suffer from collapse.

4 Experiments with CIFAR-10 and CIFAR-100

We benchmark our self-supervised learning algorithm, SSL-EY, against Barlow Twins and VICReg on CIFAR-10 and CIFAR-100 [15]. Each dataset contains 60,000 labeled images, distributed over 10 classes for CIFAR-10 and 100 classes for CIFAR-100.

We adopt a standard experimental design as detailed in [21]. We employ the sololearn library [10], which provides optimized setups for several SSL algorithms, including VICReg and Barlow Twins. All methods utilize a ResNet-18 encoder and a bi-layer projector network with 2048 hidden units and 2048 output units. Training is conducted over 1,000 epochs with batches of 256 images. For SSL-EY, we use hyperparameters that are optimized for Barlow Twins; our aim is not to outperform but to demonstrate the robustness of our method[7]. We evaluate performance using a linear probe on the learned representations and report both Top-1 and Top-5 accuracies on the validation set. For additional details, please refer to the supplementary material C.

Competitive Performance Without Hyperparameter Tuning: SSL-EY performs competitively against Barlow Twins and VICReg, as demonstrated in Table 1. Importantly, this performance is achieved using the default hyperparameters optimized for Barlow Twins, in contrast to the heavily optimized settings employed for Barlow Twins and VICReg.

Model Convergence: The Learning curves in Figure 2 indicate that the performance variation at 1,000 epochs in table 1 mainly results from optimization noise and speed of convergence is similar.

[5] We adopt bold notation to represent matrices of samples from the distribution.

[6] With corresponding $\mathbf{Z}^{(1)\prime}, \mathbf{Z}^{(2)\prime}$

[7] This was largely due to computational constraints; we hope to explore further optimisation in future work.

Smaller Projector or None at All: One key motivation for projectors is to prevent excessive collapse of meaningful information. Because SSL-EY learns does not suffer from collapse, we had a prior that it may be more robust to projector size, and perhaps even to removing the projector altogether. For this reason, in another set of experiments, we explored varying the projector's output dimensions from 2048 to 64 and removing the projector completely while holding the encoder output size constant. Figure 3a demonstrates that SSL-EY maintains good performance even with a smaller projector, making the representations more efficient than Barlow Twins and VICReg (they contain the same amount of useful information for the classification task in much fewer dimensions). While Figure 3a shows the strong performance of Barlow Twins and VICReg at larger projector sizes for this task, we would argue that our objective is more robust to this design choice, potentially offering a more reliable choice for practitioners employing SSL to unfamiliar datasets. At the bottom of Table 1, we further highlight the efficiency of SSL-EY by showing that our model performs similarly when we have no projector (just using the a 2048 dimensional representation), suggesting that SSL-EY is less reliant on this architecture[8]. In contrast, we show in appendix B.1 that Barlow Twins and VICReg's performance drops substantially without the use of a projector.

$\mathcal{L}_{\mathrm{EY}}$ is an informative metric: Figure 3b offers two key insights. First, it shows that the EY loss, which provides an unbiased estimate of the canonical correlations of the embeddings, is closely related to classification accuracy. This suggests that maximizing canonical correlation is a promising pretext task for self-supervised learning. Second, the figure reveals that even a reduced-dimensionality projector output (64 dimensions) has not reached its full capacity by 1,000 epochs. Specifically, the sum of squared canonical correlations reaches 46, out of a maximum possible value of 64. This indicates that there is still room for further optimization, implying that SSL-EY's representations have not yet saturated their capacity for capturing meaningful information. Lastly, the evolution of the correlation, as measured by $\mathcal{L}_{\mathrm{EY}}$, offers a novel way of monitoring model training even without the need for a separate validation task like classification, and could potentially eliminate the requirement for a validation set altogether. This is a particularly interesting direction given recent work on the stepwise eigenvalue behavior of the representations in SSL models [19].

Method	CIFAR-10 Top-1	CIFAR-10 Top-5	CIFAR-100 Top-1	CIFAR-100 Top-5
Barlow Twins	**92.1**	99.73	**71.38**	**92.32**
VICReg	91.68	99.66	68.56	90.76
SSL-EY	91.43	**99.75**	67.52	90.17
SSL-EY No Proj.	90.98	99.69	65.21	88.09

Table 1: Performance comparison of SSL methods on CIFAR-10 and CIFAR-100.

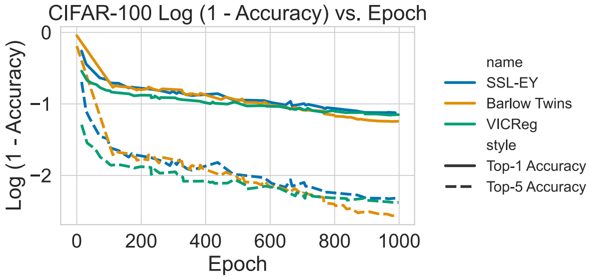
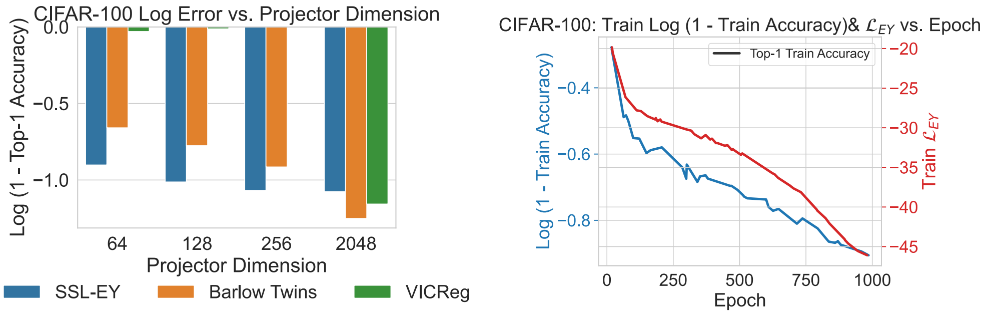


Figure 2: Learning curves for SSL-EY, Barlow Twins, and VICReg on CIFAR-100, showing performance across 1,000 epochs.

[8] We note that W-MSE, a close relative of our work, also didn't use a projector despite its use being seemingly ubiquitous



(a)

(b)

Figure 3: (a) Performance of SSL-EY with reduced projector size compared to Barlow Twins and VICReg. (b) SSL-EY's learned embeddings indicate untapped representation capacity.

5 Dicussion

In this paper, we introduced a novel self-supervised learning loss function inspired by advancements in Deep Canonical Correlation Analysis. Our method, SSL-EY, not only theoretically maximizes the correlation of learned representations but also competes favorably with existing methods like Barlow Twins and VICReg on benchmark datasets. We demonstrated that our loss function can serve as an interpretable metric for representation quality and as a stopping criterion for model training, thereby simplifying the learning process.

A downside of our approach is that in theory it requires two independent samples at each iteration to achieve unbiased updates and therefore incurs a small additional computational cost - although in practice we found that with large enough minibatches this independence could be dropped[9].

Future work will fully benchmark the performance of our proposed objective across diverse tasks and with complete tuning of optimizer parameters including learning rates. We are particularly excited by the use of different modalities and dropping the shared weights; in effect applying (deep) CCA to the increasingly available large multimodal datasets[16]. We also hope to explore the use of $\mathcal{L}_{\text{EY}}$ as a validation metric for downstream computer vision tasks using similar experiments to [1, 12].

Acknowledgments and Disclosure of Funding

JC is supported by the EPSRC-funded UCL Centre for Doctoral Training in Intelligent, Integrated Imaging in Healthcare (i4health) and the Department of Health's NIHR-funded Biomedical Research Centre at University College London Hospitals. LW is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/W524141/1.

[9] The pseudocode in Appendix 1 for example drops this assumption for ease of use within the solo-learn framework

References

[1] K. K. Agrawal, A. K. Mondal, A. Ghosh, and B. Richards. α-req: Assessing representation quality in self-supervised learning by measuring eigenspectrum decay. *Advances in Neural Information Processing Systems*, 35:17626–17638, 2022.

[2] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR, 2013.

[3] F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. 2005.

[4] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

[5] R. Balestriero and Y. LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *Advances in Neural Information Processing Systems*, 35:26671–26685, 2022.

[6] A. Bardes, J. Ponce, and Y. LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[7] M. Borga. *Learning Multidimensional Signal Processing*. PhD thesis, 1998. Publisher: Linköping University Electronic Press.

[8] X. Chang, T. Xiang, and T. M. Hospedales. Scalable and effective deep cca via soft decorrelation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1488–1497, 2018.

[9] J. Chapman, A. L. Aguila, and L. Wells. Efficient algorithms for the cca family: Unconstrained objectives with unbiased gradients, 2023.

[10] V. G. T. Da Costa, E. Fini, M. Nabi, N. Sebe, and E. Ricci. solo-learn: A library of self-supervised methods for visual representation learning. *J. Mach. Learn. Res.*, 23(56):1–6, 2022.

[11] A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024. PMLR, 2021.

[12] Q. Garrido, R. Balestriero, L. Najman, and Y. Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning*, pages 10929–10974. PMLR, 2023.

[13] ICML. ICML 2023, 2023.

[14] L. Jing, P. Vincent, Y. LeCun, and Y. Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[15] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[17] J. Ma, T. Hu, and W. Wang. Deciphering the projection head: Representation evaluation self-supervised learning. *arXiv preprint arXiv:2301.12189*, 2023.

[18] E. Sansone and R. Manhaeve. Gedi: Generative and discriminative training for self-supervised learning. *arXiv preprint arXiv:2212.13425*, 2022.

[19] J. B. Simon, M. Knutins, L. Ziyin, D. Geisz, A. J. Fetterman, and J. Albrecht. On the stepwise nature of self-supervised learning. *arXiv preprint arXiv:2303.15438*, 2023.

[20] G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. ACADEMIC PressINC, July 1990. Google-Books-ID: bIYEogEACAAJ.

[21] S. Tong, Y. Chen, Y. Ma, and Y. Lecun. Emp-ssl: Towards self-supervised learning in one training epoch. *arXiv preprint arXiv:2304.03977*, 2023.

[22] W. Wang, R. Arora, K. Livescu, and N. Srebro. Stochastic optimization for deep cca via nonlinear orthogonal iterations. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 688–695. IEEE, 2015.

[23] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

A Pseudo-Code

The version of SSL-EY in algorithm 1 is designed to integrate seamlessly into solo-learn, offering support for distributed training.

Algorithm 1: Solo-Learn Loss function for distributed SSL-EY in Python

```
# Define the SSL-EY loss function
# Input:  Projected features from two views
def SSL_EY(z1, z2):
   # Get the minibatch size and feature dimension
   N, D = z1.size()
   # Compute the covariance matrix from the concatenated features
   C = torch.cov(torch.hstack((z1, z2)).T)
   # Average the covariance matrix across all processes if distributed
    training is enabled
   if dist.is_available() and dist.is_initialized():
      dist.all_reduce(C)
      world_size = dist.get_world_size()
      C /= world_size
   # Extract symmetric and anti-symmetric blocks of C
   A = C[:D, D:] + C[D:, :D]
   B = C[:D, :D] + C[D:, D:]
   # Return the SSL-EY loss value
   return -torch.trace(2 * A - B @ B)
```

B Further Experiments and Figures

B.1 No Projector

We conducted the same experiment as in the main text for Barlow Twins and VICReg, applying the linear probe to the projector and encoder outputs. In contrast to SSL-EY, both Barlow Twins and VICReg showed a significant drop in performance when the projector was removed, as reflected in the classification metrics. Despite the similarities in the underlying motivations across all three methods, the necessity of a projector for Barlow Twins and VICReg remains an open question. Our results suggest that a projector may not be a mandatory component for correlation-based models, at least in the context of our proposed method.

		CIFAR-10		CIFAR-100	
Method	Output	Top-1	Top-5	Top-1	Top-5
Barlow Twins	Projector	92.1	99.73	71.38	92.32
Barlow Twins	No Projector	89.99	99.21	63.51	86.99
VICReg	Projector	91.68	99.66	68.56	90.76
VICReg	No Projector	**90.99**	99.46	63.82	86.39
SSL-EY	Projector	91.43	99.75	67.52	90.17
SSL-EY	No Projector	90.98	**99.69**	**65.21**	**88.09**

Table 2: SSL methods on CIFAR-10 and CIFAR-100 using 2048 unit projectors.

B.2 Robustness of our objective to Different Augmentation schemes

We used VICReg augmentations and Barlow Twins augmentations with each of our proposed methods. We found that performance was similar and therefore confirms that differences in performance across Barlow Twins, VICReg, and our two methods is driven by differences in the objective.

		CIFAR-10		CIFAR-100	
Method	Augmentation	Top-1	Top-5	Top-1	Top-5
SSL-EY	Barlow Twins	89.49	99.54	65.62	89.00
SSL-EY	VICReg	90.43	99.62	64.34	87.89

Table 3: SSL methods on CIFAR-10 and CIFAR-100 using different augmentations.

B.3 Additional Figures and Experiments on CIFAR-10

The following set of figures serves as an extension of the figures and experiments conducted on CIFAR-100, as described in the main text. These experiments on CIFAR-10 provide additional validation of our primary claims.

B.3.1 Model Convergence CIFAR-10

In Figure 4, the learning curves for SSL-EY, Barlow Twins, and VICReg on CIFAR-10 are shown. Similar to our observations in CIFAR-100, all models exhibit comparable convergence trends across 1,000 epochs.

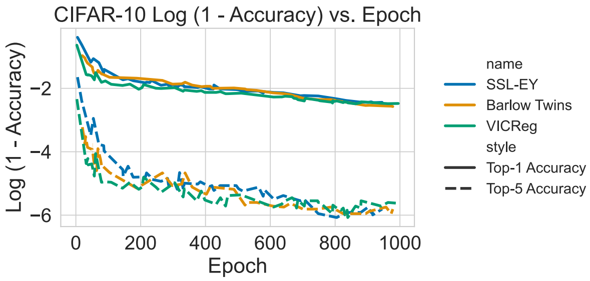


Figure 4: Learning curves for SSL-EY, Barlow Twins, and VICReg on CIFAR-10, showing performance across 1,000 epochs.

B.3.2 Projector Dimension Experiments for CIFAR-10

Figure 5 mirrors the projector dimension experiments that were initially conducted on CIFAR-100. Once again, we find that SSL-EY maintains robust performance even when the projector dimensions are reduced, contrasting with Barlow Twins and VICReg, whose performance degrades in such conditions.

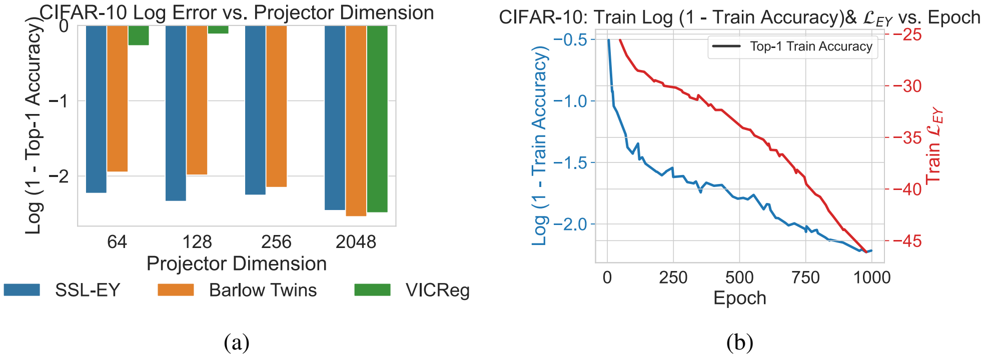


Figure 5: (a) Performance of SSL-EY with reduced projector size compared to Barlow Twins and VICReg. (b) SSL-EY's learned embeddings indicate untapped representation capacity.

C Reproducibility and Experiment Details

In this section, we provide a comprehensive overview of the experimental settings and configurations employed in our self-supervised experiments.

As previously stated, we adopt the standard setup from solo-learn's pretraining scripts. For the backbone network, we use ResNet-18. The projector network features hidden dimensions and output dimensions both set to 2048. We utilize the LARS optimizer with a learning rate of 0.3 for the backbone and 0.1 for the classifier. The batch size is configured to 256, and the weight decay is set to 1×10^{-4}. Additional optimizer parameters include setting 'clip_lr' to True, η to 0.02, and 'exclude_bias_n_norm' to True. A warmup cosine scheduler is used for learning rate scheduling. The models are trained for 1000 epochs and computations are performed with a numerical precision of 16 bits.

VICReg and Barlow Twins: Both models employ similar data augmentations, specified in Tables 4 and 5. In table 4 we show the shared augmentations while in table 5 we show the differences. Note that Barlow Twins uses two different augmentations with 50% probability each.

Augmentation	**Parameters**
ColorJitter	brightness = 0.4, contrast = 0.4, saturation = 0.2, hue = 0.1, prob = 0.8
Grayscale	prob = 0.2
HorizontalFlip	prob = 0.5
CropSize	32

Table 4: Shared augmentations for VICReg and Barlow Twins

Augmentation	**VICReg**	**Barlow Twins (crop 1)**	**Barlow Twins (crop 2)**
RandomResizedCrop	Yes	Yes	Yes
crop min scale	0.2	0.08	0.08
crop max scale	1.0	1.0	1.0
Solarization	Yes prob = 0.1	No prob = 0.0	Yes prob = 0.2
NumCrops	2	1	1

Table 5: Different augmentations for VICReg and Barlow Twins