Exploring the Gap between Collapsed & Whitened Features in Self-Supervised Learning

Bobby He [1 †] **Mete Ozay** [2]

Abstract

Avoiding feature collapse, when a Neural Network (NN) encoder maps all inputs to a constant vector, is a shared implicit desideratum of various methodological advances in self-supervised learning (SSL). To that end, whitened features have been proposed as an explicit objective to ensure uncollapsed features (Zbontar et al., 2021; Ermolov et al., 2021; Hua et al., 2021; Bardes et al., 2022). We identify power law behaviour in eigenvalue decay, parameterised by exponent $\beta \geq 0$, as a spectrum that bridges between the collapsed & whitened feature extremes. We provide theoretical & empirical evidence highlighting the factors in SSL, like projection layers & regularisation strength, that influence eigenvalue decay rate, & demonstrate that the degree of feature whitening affects generalisation, particularly in label scarce regimes. We use our insights to motivate a novel method, Post-hoc Manipulation of the Principal Axes & Trace (PostMan-Pat), which efficiently post-processes a pretrained encoder to enforce eigenvalue decay rate with power law exponent β, & find that PostMan-Pat delivers improved label efficiency and transferability across a range of SSL methods and encoder architectures.

1. Introduction

As label procurement can be expensive relative to the availability of unlabelled data, self-supervised learning (SSL), where a learning algorithm operates without access to labels, has grown both in importance & interest in recent years. Without labels, a general recipe that has produced impressive results is: 1) learning NN features/representations that are invariant to transformations of the same input, whilst 2) avoiding a completely collapsed representation, when all inputs map to a constant feature vector.

A variety of approaches have been proposed to successfully avoid feature collapse, including: contrastive learning (Chen et al., 2020a; He et al., 2020b); clustering (Caron et al., 2018; 2020); non-contrastive learning (Grill et al., 2020; Chen & He, 2021); and kernel dependence maximisation (Li et al., 2021). Of particular relevance to this work are feature decorrelation/whitening SSL methods (Ermolov et al., 2021; Zbontar et al., 2021; Hua et al., 2021; Bardes et al., 2022), which promote whitened/decorrelated features as a sufficient condition to avoid collapse.

Existing theoretical analyses into feature collapse & its mechanisms in SSL have focused on explaining why it does not occur in non-contrastive SSL (Tian et al., 2021; Zhang et al., 2022) or how a related notion of *dimensional collapse* (Hua et al., 2021), where features span a low-dimension subspace of the entire feature space, occurs (Jing et al., 2021). In these works, (dimensional) feature collapse can be interpreted in terms of a binary outcome for each dimension of the encoder NN: collapsed or uncollapsed. Thus, the size of uncollapsed feature dimensions and the importance of their rate of decay have so far been unexplored in SSL.

In this work, we examine the gap between collapsed & whitened features, highlighting its significance by first identifying power law behaviour in eigenvalue decay as a bridge between collapse & whitening in Section 2. We theoretically & empirically study elements of SSL that affect eigenvalue decay rate in Section 3, including projector head depth & regularisation strength, before demonstrating that generalisation performance in SSL is non-monotonic in the degree of feature whitening/collapse in Section 4. We show that the extent of feature whitening has implications for generalisation in low-labelled data regimes in Section 4.1, & use this to motivate our methodological contribution in Section 5: Post-hoc Manipulation of the Principal Axes & Trace (PostMan-Pat or PMP), which takes a pretrained SSL encoder and enforces a power law in its feature eigenspectrum. In Section 6, we show that PMP improves label-efficiency and transferability of pretrained SSL encoders under linear evaluation & often outperforms semi-supervised finetuning. For example, a pretrained Barlow Twins (Zbontar et al., 2021) encoder is improved by over 1% top-1 accuracy (56.2% vs 55.0%) on ImageNet-1K with only 1% of labels.

[†]Researched while interning at Samsung Research UK. [1]University of Oxford, [2]Samsung Research UK. Correspondence to: Bobby He <bobby.he@stats.ox.ac.uk>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

2. Background, Related Work, & Notation

Suppose we have a large unlabelled dataset $\boldsymbol{X}=\{\boldsymbol{x}_n\}_{n=1}^N$, with N samples of dimension d. We denote the empirical distribution over $\boldsymbol{X}$ by $\hat{p}(\boldsymbol{x})$. The SSL setting we consider is to learn a useful feature encoder NN, $h_{\boldsymbol{\theta}}$: $\mathbb{R}^d \to \mathbb{R}^{d_e}$ from $\boldsymbol{X}$, with trainable parameters $\boldsymbol{\theta}$, for downstream tasks. If appropriate, we drop the $\boldsymbol{\theta}$ subscript for clarity.

Self-supervised representations are generally evaluated with a labelled/supervised dataset of samples $\boldsymbol{X}_S \in \mathbb{R}^{S\times d}$ and labels $\boldsymbol{Y}_S \in \mathbb{R}^{S\times C}$. We assume $\boldsymbol{X}_S$ are sampled i.i.d. from the same marginal distribution $p(\boldsymbol{x})$ as our unlabelled $\boldsymbol{X}$, and that there is an additional conditional $q(\cdot|\boldsymbol{x})$ such that $\boldsymbol{y}_s|\boldsymbol{x}_s \sim q(\boldsymbol{y}|\boldsymbol{x}_s)$. Typically the task is C-class classification, & a linear layer $W_C \in \mathbb{R}^{d_e\times C}$ is composed on top of the encoder $h_{\boldsymbol{\theta}}$ to give predictor $f(\boldsymbol{x})=h_{\boldsymbol{\theta}}(\boldsymbol{x})W_C$.

Evaluation is usually via: 1) non-linear finetuning by training both W_C & $h_{\boldsymbol{\theta}}$ (*finetuning*) or, 2) linear training of W_C only (*linear probe*). Linear probes train the following loss, where l is typically cross-entropy & $\frac{\sigma^2}{S}$ is weight decay:

$$\mathcal{L}(W_C) = \sum_{s=1}^{S} l(f(\boldsymbol{x}_s), \boldsymbol{y}_s) + \sigma^2 \|W_C\|_2^2. \qquad (1)$$

As mentioned, recent approaches (Chen et al., 2020a; Caron et al., 2020; Zbontar et al., 2021; Grill et al., 2020) all adopt the idea of training $\boldsymbol{\theta}$ to be invariant to a distribution of $\mathbb{R}^d \to \mathbb{R}^d$ transformations $\mathcal{T}$ that preserves semantic content (like random crops). In other words, given an image $\boldsymbol{x}\in\mathbb{R}^d$, we have $h_{\boldsymbol{\theta}}(T_1(\boldsymbol{x})) \approx h_{\boldsymbol{\theta}}(T_2(\boldsymbol{x}))$ for $T_1, T_2 \overset{\text{i.i.d.}}{\sim} \mathcal{T}$, where the joint-embeddings $T_1(\boldsymbol{x})$ & $T_2(\boldsymbol{x})$ are known as a *positive pair*. Figure 1 visualises this general approach.

Where these methods differ is in how they avoid the trivial (& useless) solution of *collapsed features*: $h_{\boldsymbol{\theta}}(\boldsymbol{x}') \triangleq \mathbf{c}$, $\forall \boldsymbol{x}' \in \mathbb{R}^d$, for constant $\mathbf{c} \in \mathbb{R}^{d_e}$. We highlight two popular approaches to avoid collapse:

Contrastive SSL methods avoid collapse by simultaneously encouraging representations of different images (*negative pairs*) $\boldsymbol{x} \neq \boldsymbol{x}'$ to be further apart in contrast to positive pairs' representations through the InfoNCE loss (Oord et al., 2018). SimCLR (Chen et al., 2020a;b) demonstrates the benefit of scaling to large batch sizes for contrastive SSL & introduces several techniques that seem to improve SSL performance in practice, such as stronger data augmentation, and the use of trainable (nonlinear) MLP projector heads $g(\cdot) : \mathbb{R}^{d_e} \to \mathbb{R}^{d_p}$. The projection $g(\cdot)$ takes encoder outputs $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ as input, so the InfoNCE loss actually acts on projections $z(\boldsymbol{x}) \triangleq g(h_{\boldsymbol{\theta}}(\boldsymbol{x}))$, not encodings $h_{\boldsymbol{\theta}}(\boldsymbol{x})$. It has been suggested that projectors serve to prevent encoder dimensional collapse (Jing et al., 2021) & ease the encoder's constraints on transformation invariance (Bordes et al., 2021). We see in Section 3 that another effect of projectors could be to whiten representations. MoCo (He et al., 2020b; Chen et al., 2020c) uses a memory bank to ease the large batch size bottleneck of contrastive SSL.

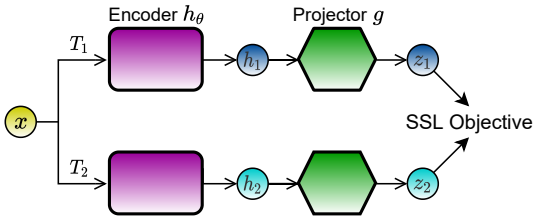

Figure 1. Schematic of joint-embedding approach in SSL.

Feature decorrelation SSL removes the need for negative pairs (hence large batch size) instead by encouraging whitened/decorrelated projections $z(\boldsymbol{x})$ over $\boldsymbol{x} \sim \hat{p}(\boldsymbol{x})$, as a sufficient condition to avoid feature collapse. W-MSE (Ermolov et al., 2021) uses an explicit Cholesky transformation to enforce an exactly whitened representation, by which we mean that the empirical distribution of representations $\{z(\boldsymbol{x}_n)\}_{n=1}^N$ has mean $\mathbf{0}$ & identity covariance $\Sigma \triangleq \frac{1}{N} z(\boldsymbol{X})^T z(\boldsymbol{X}) = I_{d_p \times d_p}$. Hua et al. (2021) considered *dimensional collapse*, where Σ has $m < d_p$ non-zero eigenvalues, as a milder but also undesirable form of collapse, thus motivating their study of feature whitening SSL.

We see that the degree of (projection) feature whitening or collapse can be defined through the eigenvalues of the covariance matrix Σ: identical non-zero eigenvalues give entirely whitened representations, whereas a collapsed representation has all-zero eigenvalues (or a single non-zero eigenvalue if the representation is uncentred). Between these two extremes there is a spectrum of possibilities for how the eigenvalues $\{\lambda_i\}_{i=1}^{d_p}$ of covariance Σ decay, which we characterise as follows using power law behaviour:

Definition 2.1 (β-power law of eigenvalues)**.** Let $\boldsymbol{\phi}(\boldsymbol{X}) \in \mathbb{R}^{N\times d_\phi}$ be a representation of $\boldsymbol{X}$ with feature-wise covariance matrix $\Sigma^\phi \in \mathbb{R}^{d_\phi \times d_\phi}$ and corresponding sorted eigenvalues $\{\lambda_i\}_{i=1}^{d_\phi}$. We say $\boldsymbol{\phi}(\boldsymbol{X})$ follows an *eigenvalue power law with exponent* $\beta \geq 0$ if there exist[1] $\Theta_{d_\phi}(1)$ positive constants $a \leq b$, such that $\forall i \geq 1$: $\frac{a}{i^\beta} \leq \lambda_i \leq \frac{b}{i^\beta}$.

Definition 2.2 (Whitened & collapsed representation)**.** We say a representation $\boldsymbol{\phi}(\boldsymbol{X})$ is *whitened* if it follows an eigenvalue power law with exponent $\beta = 0$, & *collapsed* if it follows an eigenvalue power law with exponent $\beta = \infty$.

Remark 2.3. Def. 2.1 is a softer definition than one where the eigenvalues follow a strict power law (i.e. $a = b > 0$), and is found elsewhere in the literature, e.g. Jin et al. (2021).

In Def. 2.1, $\boldsymbol{\phi}(\boldsymbol{X})$ can be any function of $\boldsymbol{X}$. At various points, we consider: the identity function; a hidden layer of

[1] We presume d_ϕ to be a large but finite feature dimension; in an NN d_ϕ corresponds to the NN width.

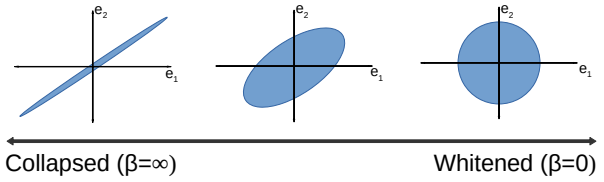

Figure 2. Toy illustration of the gap between collapsed & whitened features as two extremes of a spectrum. The 2D feature covariances above share eigenvectors, but have different eigenvalues.

an encoder h; an encoder h; or an encoder h + projection g. Figure 2 depicts a 2D visualisation of the range between collapse and whitened features, which may be parameterised by power-law exponent β in eigenvalue decay, as in Def. 2.1.

In lieu of explicit whitening, Barlow Twins (Zbontar et al., 2021) introduces a regularised loss $\mathcal{L}_{\text{BT}}$, with strength $\rho{>}0$ providing a soft constraint on feature correlation:

$$\mathcal{L}_{\text{BT}} = \sum_{i=1}^{d_p} \left[(1 - C_{ii})^2 + \rho \sum_{j \neq i} C_{ij}^2 \right], \tag{2}$$

where for $\boldsymbol{x}{\sim}\hat{p}(\boldsymbol{x})$ & $\{T_a\}_{a\in[2]} \overset{\text{i.i.d.}}{\sim} \mathcal{T}$, we have correlations:

$$C_{ij} = \frac{\mathbb{E}\big[\bar{z}_i(T_1(\boldsymbol{x}))\bar{z}_j(T_2(\boldsymbol{x}))\big]}{\mathbb{E}\big[\bar{z}_i(T_1(\boldsymbol{x}))^2\big] \cdot \mathbb{E}\big[\bar{z}_j(T_2(\boldsymbol{x}))^2\big]}, \tag{3}$$

with $\bar{z}_i(\boldsymbol{x}) \triangleq z_i(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}\sim\hat{p}}[z_i(\boldsymbol{x})]$ defined to be centred. In practice, the expectations in Eq. (3) are estimated with mini-batches & random transformations sampled from $\mathcal{T}$. Note that, up to input transformations, we have $C_{ij}{=}\frac{\Sigma^z_{i,j}}{\sqrt{\Sigma^z_{i,i}\Sigma^z_{j,j}}}$.

In $\mathcal{L}_{\text{BT}}$, the on-diagonal elements of C encourage $z(\boldsymbol{x})$ to be invariant to transformations of the same image, whereas the off-diagonal contributions encourage the d_p individual projection features to be pairwise uncorrelated across inputs, preventing collapse (of both the encoder and projection).

Key takeaways: β-power law of eigenvalues in SSL.

- Power law behaviour, determined by exponent $\beta > 0$, in feature eigenspectrum decay is one *possible* way to bridge the gap between collapsed & whitened features.

3. Projection Layers Whiten Eigenspectra

Having highlighted the gap between collapsed and whitened features, we now study the factors, such as projector layers and choice of SSL method, that influence where a pretrained encoder lies along this spectrum.

Intuitively, low Barlow Twins training loss results in whitened projections, if individual neurons' variance across inputs satisfies $\Sigma^z_{ii} = \Theta(1)$ for $i \in [d_p]$. This is because if $\mathcal{L}_{\text{BT}}$ is small, then Σ^z is approximately diagonal, and we can read off the eigenvalues $\{\Sigma^z_{ii}\}_i$, as follows:

Proposition 3.1. *Suppose an NN encoder+projection trained via Barlow Twins (Zbontar et al., 2021) achieves (i) training loss $\mathcal{L}_{BT} = \epsilon$, & ii) $\exists a \leq b$ positive constants such that $a \leq \Sigma^z_{ii} \leq b \ , \forall i{\in}[d_p]$. Then, if $\epsilon \leq \frac{a^2\rho}{b^2}$, the projector has a whitened eigenspectrum.*

From Proposition 3.1 (proof in Appendix A), a successfully trained Barlow Twin encoder obtains a whitened projection eigenspectrum. Moreover, the larger the regularisation strength ρ, the more likely the condition $\epsilon \leq \frac{a^2\rho}{b^2}$ is to be satisfied, so larger ρ leads to more whitened projector eigenspectra, which we later empirically confirm in Figure 4.

To justify our assumptions on feature variances, we note that ensuring $\Sigma^z_{ii}{=}\Theta(1)$, $\forall i{\in}[d_p]$, is one motivation for the VICReg (Bardes et al., 2022) extension of the Barlow Twins loss function $\mathcal{L}_{\text{BT}}$. Having said that, we empirically verify Proposition 3.1 in Figure 3 (center left) for a Barlow Twin ResNet-18 on CIFAR-10, where we plot (normalised) projection eigenvalues by size as training progresses.

As seen in Figure 3, at initialisation, the projection eigenspectrum is dominated by one eigenvalue. Through training, the relative size of smaller eigenvalues increases, such that after 100 epochs, we have around 120 dominant eigenvalues within an order of magnitude of the largest eigenvalue. We note some dimensional collapse is still observed for Barlow Twins projectors, as found for BYOL (Grill et al., 2020; Tian et al., 2021) & SimCLR (Chen et al., 2020a; Jing et al., 2021), as the encoder dimension is 512 with projector width of 1024. We leave an exploration of dimensional collapse in feature decorrelation methods like Barlow Twins for future work, & focus here on the decay rate of the dominant eigenvalues, both for Barlow Twins and in general in SSL.

This whitened eigenspectrum property of feature decorrelation SSL projections is somewhat at odds with findings from neuroscience (Stringer et al., 2019), where it has been observed empirically that neuronal population responses in the visual cortex of mice follow an eigenspectrum power law decay with $\beta{=}1$ in Def. 2.1. In all subplots of Figure 3, we plot the line $y = \frac{1}{x}$ for reference, observing that the dominant projection eigenvalues decay much slower than a $\beta{=}1$ power law. However, in Figure 3 (left) we also see that the eigenvalues for the corresponding encoder decay much faster compared to the projector (noting the log-log scale).

To provide theoretical support for this observation that projection layers change the eigenspectra to encourage faster eigenspectrum decay in encoder layers, we consider the setting of a deep linear MLP, with widths d_l at layer l satisfying $d_l > d \ , \forall l \in [L]$ (so that the MLP is wider than input dimension). Under an additional assumption of alignment between the first layer weight matrix $W_1 \in \mathbb{R}^{d_1 \times d}$ and the

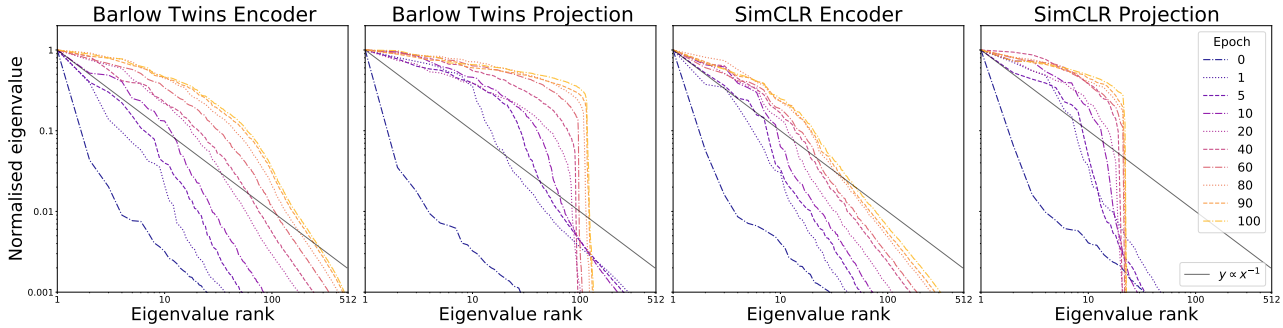

Figure 3. Eigenspectra for features & projections of both Barlow Twins & SimCLR networks. Projector MLPs use ReLU & have depth 2.

input covariance matrix $\Sigma^{\boldsymbol{x}} \in \mathbb{R}^{d\times d}$, we show that changes in the eigenspectrum decay between input and output layers are evenly spaced amongst hidden layers in Corollary 3.4.

Definition 3.2. $A{\in}\mathbb{R}^{a\times c}$ & $B{\in}\mathbb{R}^{c\times b}$ are *aligned* if there exist Singular Value Decompositions (SVDs) $A{=}U_A D_A V_A^T$ and $B{=}U_B D_B V_B^T$ such that $V_A^T U_B{=}I_{c\times c}$.

We note that this input-layer alignment phenomenon has been shown for the top principal component for linear NNs (Ji & Telgarsky, 2018), has been proven & observed empirically for deep non-linear NNs trained on random labels (Maennel et al., 2020), & that first layer alignment has been exploited to design Bayesian NN priors that are robust to covariate shift (Izmailov et al., 2021). Adjacent-layer alignment has been shown for contrastive SSL (Jing et al., 2021).

Proposition 3.3. *Suppose we have an L-layer linear MLP, $f(\boldsymbol{x}){=}\prod_{l=1}^{L} W_l \cdot \boldsymbol{x} \in \mathbb{R}^{d_L}$, trained to convergence using gradient flow & no bias terms on some loss $\mathcal{L}(f(\boldsymbol{X}))$ with weight decay $\eta > 0$. Assume further that the first layer matrix $W_1 \in \mathbb{R}^{d_1\times d}$ & input covariance matrix $\Sigma^{\boldsymbol{x}} \in \mathbb{R}^{d\times d}$ are aligned as in Def. 3.2. Then:*

1. *Adjacent layers' matrices W_l & W_{l-1} become aligned during training for $1 < l \leq L$ so that principal components can be grouped together across layers. If $\lambda_{l,j}$ denotes the j^{th} eigenvalue of the empirical covariance of features at layer l, then:*
2. *For any uncollapsed output eigenvalue j with $\lambda_{L,j}{>}0$ & any two layers $0{\leq}k{<}l$ (where $k = 0$ denotes the input layer), we have:*

$$\lambda_{l,j}{=}(\lambda_{k,j})^{\frac{L-l}{L-k}}(\lambda_{L,j})^{\frac{l-k}{L-k}},$$

i.e. $\lambda_{l,j}$ is a weighted geometric mean between $\lambda_{k,j}$ & $\lambda_{L,j}$, with weighting specified by closeness to k or L.

The proof of Proposition 3.3 can be found in Appendix A, & is largely inspired from previous work (Saxe et al., 2013; Ji & Telgarsky, 2018; Tian et al., 2021; Jing et al., 2021). However, it allows us to deduce that deeper projection MLPs result in a faster decaying encoder eigenspectrum:

Corollary 3.4. *In the setting of Proposition 3.3, suppose we have a fixed encoder depth l_e, & that for some encoder layer $k < l_e$, the feature eigenspectrum at layer k follows power law decay with exponent $\beta > 0$. Then, deeper projection MLPs result in faster encoder eigenvalue decay, if projection outputs are whitened (as in Proposition 3.1).*

Proof. Suppose the projector has depth l_p giving combined depth $L{=}l_e{+}l_p$. We are given $\lambda_{k,j}{=}j^{-\beta}$ & $\lambda_{L,j}{=}1$ up to constant, $\forall j$ (for simplicity of argument here we suppose $a = b$ in Def. 2.1). Applying Proposition 3.3 at the encoder layer, we conclude $\lambda_{l_e,j}{=}j^{-\frac{l_p}{L-k}\beta}$, i.e. power law behaviour with exponent $\frac{l_p\beta}{l_p+l_e-k}$, which is increasing in l_p. □

Remark 3.5. We use power law behaviour in Corollary 3.4 as a convenient medium to express eigenspectrum decay rate, though our conclusion extends to representations without an obvious eigenvalue power law (c.f. Figures 3 and 4).

We now justify our assumptions on eigenspectra in Corollary 3.4. For hidden layers, there are at least two phenomena that encourage hidden feature eigenspectra to decay in practice: 1) the fact that, at the input layer, natural images inherently possess such a power law decaying eigenspectrum (Field, 1987; Ruderman & Bialek, 1994) and 2) it is well known in the signal propagation/wide NN literature that common NN initialisation schemes (e.g. Kaiming (He et al., 2015)) converge to collapsed representations (with an at best polynomial rate) in depth (Schoenholz et al., 2016; Hayou et al., 2019; 2021; Martens et al., 2021). This is corroborated by Figure 3, where both projection & encoders' eigenspectra are dominated by the largest eigenvalue at epoch 0.

For the output eigenspectrum, decorrelation SSL methods are covered by Proposition 3.1 and Figure 3. However, we note that Proposition 3.3 and Corollary 3.4 are agnostic to the specific loss function $\mathcal{L}$ & can also apply to SimCLR with InfoNCE loss (Oord et al., 2018). In Figure 3, we empirically observe for a SimCLR-trained NN that the biggest normalised eigenvalues for projection features (right) are

larger compared to the encoder eigenspectrum (center right), although the effect is weaker for SimCLR compared to Barlow Twins, perhaps due to increased dimensional collapse in the projector (Hua et al., 2021; Jing et al., 2021).

Key takeaways: Whitening and collapse in SSL.

- Trained Barlow Twins NNs have whitened projections.
- Whitened projections don't imply whitened encoders, especially for non-whitened inputs & deeper encoders.
- Deeper projector MLPs may result in more collapsed encoder eigenspectra (empirically verified in Figure 4).

4. Whitened Features Affect Generalisation

In the previous section, we've seen theoretical & empirical evidence that interactions exist between SSL design choices (e.g. projection depth, or choice of method), and the speed of eigenspectrum decay of encoder features. We now demonstrate the rate of decay in the encoder eigenspectrum is important for the quality of learnt SSL representations, in terms of generalisation under linear evaluation. In particular, we show that the relationship between degree of feature whitening & generalisation is not monotonically increasing.

We first seek a quantitative metric to measure how whitened a feature representation is, beyond the power law exponent β of Def. 2.1, as we wish to handle representations that do not possess an obvious eigenvalue power law. Instead, we consider the *normalised eigenvalue sum* (NESum):

Definition 4.1. Given a feature representation $\boldsymbol{\phi}(\boldsymbol{X}) \in \mathbb{R}^{N \times d_\phi}$ with feature-wise covariance matrix $\Sigma \in \mathbb{R}^{d_\phi \times d_\phi}$ and eigenvalues $\{\lambda_i\}_{i=1}^{d_\phi}$ in decreasing order. Then, we define the *normalised eigenvalue sum* to be:

$$\text{NESum}(\{\lambda_i\}_i) \triangleq \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_1}$$

with convention $\frac{0}{0}=0$. NESum takes values in $[0, d_\phi]$, with collapsed features having NESum$=0$, and NESum$=d_\phi$ corresponding to exactly whitened features.

Moreover, it is clear via a geometric series argument that for power law decaying feature eigenspectra with exponent β, NESum decreases as β increases, so we view larger values of NESum to denote whiter representations.

In Figure 4, we compare various ResNet-18 trained with Barlow Twins on CIFAR-10, in terms of encoder NESum against test accuracy under linear probe. For projection MLP depths from 1 to 5 & a range of regularisation strengths ρ (on a logarithmic scale from 0.001 to 0.05), we plot markers for NNs trained from three independent initialisations. The different coloured lines denote splines interpolating the three-seed averages across ρ for different depths. The size of the markers correspond to regularisation strength ρ.

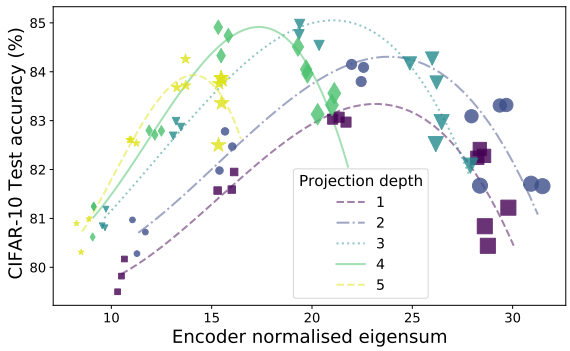

Figure 4. Different Barlow Twins networks on CIFAR-10 with ResNet18 encoder. Each marker corresponds to a trained NN, with marker size denoting regularisation strength ρ. Different coloured lines are spline fits split by projection MLP depth. We see that test accuracy across all depths does not monotonically increase as features become less collapsed.

As suggested by Corollary 3.4, we see that the deeper the projection MLP, the lower NESum across different values of ρ, indicating that encoder NNs trained with shallower projectors are more whitened. Moreover, for each depth, we see that too low NESum results in lower test accuracy, presumably because the features are too collapsed as expected from previous works studying collapse in SSL.

On the other hand, we also see that too high a value of NESum (which we see is due to larger ρ from the marker sizes, as supported by Proposition 3.1) results in lower test accuracy too. This might seem perplexing from the existing feature decorrelation SSL literature where whitening is simply used as a mechanism to avoid feature collapse, and so one might expect a monotonically increasing relationship between NESum & test accuracy.

However, we note that this is somewhat unsurprising given that eigenspectra in biological NNs such as a mouse's visual cortex are known to decay (Stringer et al., 2019). Thus, Figure 4 suggests that in SSL, we should not only seek to avoid too collapsed feature representations, but also too whitened representations. A corresponding figure with STL10 dataset can be found in Appendix B, with similar conclusions.

Key takeaways: Whitening and generalisation in SSL.

- The relationship between extent of feature whitening & generalisation performance in SSL is not monotonic.
- Lower Barlow Twins regulariser ρ yields more collapsed features.

4.1. Insights for Generalisation on Low Labelled Data

To examine the impact of feature whitening in SSL theoretically, we turn to the setting of low labelled data, when $S \ll N$ in the notation of Section 2. We focus on linear evaluation, as opposed to semi-supervised finetuning. Notwithstanding the fact that linear evaluation is one of the

main benchmarks for evaluating SSL methods, nonlinear finetuning in the setting of small labelled-data is vulnerable to tampering with useful features acquired from large-scale unlabelled data. Empirically, we demonstrate this in Section 6 on small labelled-data ImageNet evaluation, where we show that linear evaluation schemes can outperform semi-supervised non-linear finetuning.

Moreover, linear evaluation of SSL lends itself more kindly to theoretical analysis, particularly connections to kernel methods/Gaussian processes (GPs) (Sollich, 1999; Sollich & Halees, 2002; Sollich, 2002; Bordelon et al., 2020; Jin et al., 2021; Cui et al., 2021). These works study *learning curves* of kernel/GP regression, describing how generalisation error changes with the amount of labelled data S.

To this end, we define a kernel $k(\boldsymbol{x},\boldsymbol{x})=\langle h_{\boldsymbol{\theta}}(\boldsymbol{x}), h_{\boldsymbol{\theta}}(\boldsymbol{x}')\rangle$ using the inner product of encoder features. If inputs are assumed to have compact support e.g. normalised, then we can use Mercer's Theorem (Mercer, 1909) to decompose k:

$$k(\boldsymbol{x},\boldsymbol{x}') = \sum_{i=1}^{d_e} \lambda_i \psi_i(\boldsymbol{x})\psi_i(\boldsymbol{x}'), \tag{4}$$

with kernel eigenfunctions $\{\psi_i\}_i$ & eigenvalues $\{\lambda_i\}_i$ (equivalent to the covariance eigenvalues we consider in Def. 2.1) satisfying $\int k(\boldsymbol{x},\boldsymbol{x}')\psi_i(\boldsymbol{x}')p(\boldsymbol{x}')d\boldsymbol{x}' = \lambda_i\psi_i(\boldsymbol{x})$.

We consider a single output $C=1$ for simplicity, as the results we use extend straightforwardly for $C>1$ (Bordelon et al., 2020). We also consider squared error instead of cross-entropy, noting that solving classification tasks with squared error (by treating labels as one-hot regression targets) is often used due to the connection with kernels (Lee et al., 2019; Shankar et al., 2020; Lee et al., 2020; He et al., 2020a). In this case, we obtain trained predictor $\hat{f}_S(\boldsymbol{x})=k(\boldsymbol{x},\boldsymbol{X}_S)\big(K_{\boldsymbol{X}_S,\boldsymbol{X}_S}+\sigma^2 I_S\big)^{-1}\boldsymbol{Y}_S$ from Eq. (1), where $K_{\boldsymbol{X}_S,\boldsymbol{X}_S}\in\mathbb{R}^{S\times S}$ is the Gram matrix of $h(\boldsymbol{X}_S)$. Finally, let us assume we have noiseless observations $q(\boldsymbol{y}|\boldsymbol{x})=\delta_{f^*(\boldsymbol{x})}(\boldsymbol{y})$ & the true target function f^* satisfies:

$$f^*(\boldsymbol{x}) = \sum_{i=1}^{d_e} \mu_i \psi_i(\boldsymbol{x}). \tag{5}$$

Then, existing works have derived the learning curves for generalisation error of $\hat{f}_S$, $\hat{\mathcal{E}}_S=\mathbb{E}_{p(\boldsymbol{x})}[\big(\hat{f}(\boldsymbol{x})-f^*(\boldsymbol{x})\big)^2]$, when k & f^* both observe power law behaviour:

Proposition 4.2 (Bordelon et al. (2020); Jin et al. (2021))**.** *If $\lambda_i=\Theta(i^{-\beta})$ & $\mu_i^2=\Theta(i^{-\alpha})$, $\forall i$, with $\alpha>1$ to ensure f^* square integrable, then as $S\to\infty$, we have $\hat{\mathcal{E}}_S=\Theta(S^{\frac{1-\alpha}{\beta}})$.*

We see in Proposition 4.2 that if $\{\lambda_i\}_i$ & $\{\mu_i\}_i$ observe power laws, then so does the generalisation error $\hat{\mathcal{E}}_S$. Moreover, for fixed f^* (& $\alpha > 1$), larger β results in slower decaying error as S increases. This allows us to conclude:

Corollary 4.3. *Let k' be defined like k in Eq. (4) but with new eigenvalues $\{\lambda'_i\}_i$, & corresponding generalisation error $\hat{\mathcal{E}}'_S$. If $\lambda'_i \sim \Theta(i^{-\beta'})\ \forall i$, then we have $\log\frac{\hat{\mathcal{E}}_S}{\hat{\mathcal{E}}'_S} \sim c + (1-\alpha)(\beta^{-1}-\beta'^{-1})\log(S)$, for some c.*

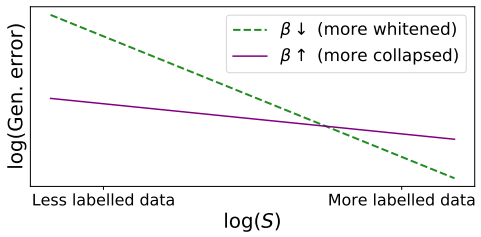

Figure 5. Power law in S for generalisation error $\hat{\mathcal{E}}_S$.

Figure 5 illustrates Corollary 4.3's implications: kernels with small β enjoy fast decaying error curves as $S\to\infty$, but this means for low S, the relative error is larger for small β. In Figure 5, we see that a more collapsed feature eigenspectrum can perform better at low labelled data S, despite performing worse at larger S. Though hypothetical, Figure 5 shows that one can alter generalisation by simply regulating eigenvalue decay, which we utilise in Section 5.

In Appendix A, we prove Theorem A.2, which shows that kernels with whitened eigenspectra may perform relatively worse at low S, compared to unwhitened eigenspectra, for f^* without power law assumptions.

Key takeaways: Generalisation and labelled data size.

- Decay rate in feature eigenspectra affects how generalisation changes with labelled data size.
- More collapsed features may perform relatively better than more whitened features on small labelled data.

5. PostMan-Pat: Post-hoc Manipulation of the Principal Axes & Trace

In the previous section, we have seen that the degree to which features are whitened has an impact on generalisation error, particularly when one has varying amounts of labelled data S. The motivation for our methodological contribution is now clear: one can affect (& potentially improve) the generalisation performance of SSL simply by explicitly controlling the eigenspectrum decay of trained SSL methods.

Our method, named Post-hoc Manipulation of the Principal Axes & Trace or PostMan-Pat (PMP), rescales the principal components of *any* encoder's covariance Σ_h *after pretraining* to enforce a power law decay in the eigenvalues $\{\lambda_i\}_i$, with exponent β acting as a hyperparameter. We do so efficiently by estimating the encoder covariance matrix $\Sigma_h \in \mathbb{R}^{d_e\times d_e}$ using our unlabelled data $\boldsymbol{X}$, from which

we derive an untrainable rescaling matrix $W_{\text{PMP}} \in \mathbb{R}^{d_e \times d_e}$, detailed in Alg. 1. Then, we redefine our pretrained encoder by appending W_{PMP}, $h_{\text{PMP}}(\boldsymbol{x}) \leftarrow h_{\boldsymbol{\theta}}(\boldsymbol{x}) W_{\text{PMP}}$, before training linear classifier W_C in Eq. (1) as before. Pseudocode for PMP is provided in Alg. 1.

Algorithm 1 PyTorch pseudocode for PostMan-Pat (PMP).

```
# h: Pretrained encoder (with standardised neurons).
# beta: Power law exponent.
# base_rank: Rank from which to start power law.
# B: Batch size.
# N: Unlabelled data size.
# D: Dimensionality of the encoder embeddings.
# harmonic: D-dim tensor with the i^th element 1/(i+1).
# mm: Matrix-matrix multiplication.
# eig: SVD operator (eigenvalues in decreasing order).

class PMPEncoder(nn.Module):
    def __init__(self, encoder, W_pmp):
        super().__init__()
        self.encoder = encoder
        W_pmp.requires_grad = False # Fixed
        self.W_pmp = W_pmp

    def forward(self, x):
        x = self.encoder(x)       # 1xD
        return mm(x, self.W_pmp) # 1xD

# Compute feature covariance matrix.
cov = torch.zeros(D, D)
for x in loader:
    z = h(x)                  # BxD
    cov += mm(z.T, z) / N # DxD

# Compute W_pmp.
eig_vals, eig_vecs = eig(cov)
eig_ratio = eig_vals[base_rank] / eig_vals[base_rank:]
eig_rescaled = torch.ones(D)
eig_rescaled[base_rank:] = eig_ratio * \
        (base_rank * harmonic[base_rank:]).pow(beta)

W_pmp_sqrt = eig_vecs * eig_rescaled.sqrt() # DxD
W_pmp = mm(W_pmp_sqrt, eig_vecs.T)          # DxD

# New PMP encoder for linear evaluation in Eq.(1).
h_pmp = PMPEncoder(h, W_pmp)
```

We next show that PMP does indeed result in a power law decaying eigenspectra with exponent β:

Proposition 5.1. *The PMP encoder h_{PMP}, as described in Alg. 1, has i) a β-power law eigenspectrum, & ii) the same left & right eigenvectors as h.*

Proof. Let $UD^{\frac{1}{2}}V^T$ be SVD of $h(\boldsymbol{X})$, so that $\Sigma_h = \frac{1}{N} VDV^T$. We defined $W_{\text{PMP}} \triangleq VR^{\frac{1}{2}}V^T$ in Alg. 1, for R diagonal & $R_{ii} = \frac{D_{rr}}{D_{ii}} (\frac{r}{i})^{\beta}$ if $i \geq r$ & 1 else, where r denotes the base rank.

So if $h_{\text{PMP}}(\boldsymbol{x}) = h(\boldsymbol{x}) W_{\text{PMP}}$, then $h_{\text{PMP}}(\boldsymbol{X}) = h(\boldsymbol{X}) W_{\text{PMP}}$ has SVD: $UD_{\text{PMP}}^{\frac{1}{2}} V^T$ where D_{PMP} satisfies $\left(D_{\text{PMP}}\right)_{ii} = D_{rr} (\frac{r}{i})^{\beta}$ if $i \geq r$ & D_{ii} else. □

As ℓ_2-regularisation/weight decay can be thought of as minimising an unregularised version of Eq. (1), $\sum_{s=1}^{S} l(f(\boldsymbol{x}_s), \boldsymbol{y}_s)$, subject to constraints on the ℓ_2-norm of W_C, we see that PMP can be viewed as an alternative to linear probes, but with eigendecomposition specific constraints. We speculate this specificity allows PMP to outperform linear probes on complex datasets, like ImageNet in Section 6.

PMP can also be viewed as using a 2-layer linear MLP classifier, instead of W_C in Eq. (1), but with fixed first layer. Training both layers in a 2-layer MLP adds non-convexity to the optimisation process: we compare PMP to a 2-layer linear MLP classifier with all layers trainable in Section 6.

Implementation details PMP requires a single eigendecomposition of Σ_h, which is 2048×2048 for a ResNet-50, so adds minimal cost to the standard ImageNet benchmark for SSL. To compute Σ_h, we need a single forward pass over unlabelled data $\boldsymbol{X}$, or we can estimate Σ_h with a moving average online. We standardise the d_e neurons in h to have zero mean and unit variance to avoid non-zero means resulting in a dominant largest eigenvalue. Though Corollary 4.3 and Figure 5 suggest larger values of β may be preferable with smaller S, we stress there are interactions with ℓ_2 regularisation not covered by our theory (c.f. Figure 10), & it is important to tune hyperparameters (as usual for complex machine learning tasks), in PMP's case β & W_C's weight decay, for best results.

6. Experiments

In interest of space, experimental details not covered in the main paper can be found in Appendix C. In all PMP experiments we start the power-law behaviour after the tenth largest eigenvalue, in line with Nassar et al. (2020), who studied importance of power-law decay in eigenspectra for adversarial robustness in NNs.

Analysis of PMP on CIFAR-10 In Figure 6, we examine the efficacy of PMP on CIFAR-10 for Barlow Twins, SimCLR and a pretrained supervised NN compared to standard linear probe evaluation. All methods use ResNet-18 encoder. On the leftmost column, we plot eigenspectra for different values of power law exponent β, compared to the original encoder. The linear trends (after the 10th largest eigenvalue) observed on log-log scale indicate that PMP successfully induces a power law, with larger β resulting in faster decaying eigenvalues. SSL methods on the left-most column have projector depth 2.

In all other columns, we plot the relative change in test accuracy when using PMP compared to standard linear probe (on the same pretrained encoder), as a function of the power law β. Error bars indicate 95% confidence over 20 data splits (if applicable) & 3 independent encoder initialisations.

We see that the performance of PMP is monotonically increasing in β for the lowest amounts of labelled data, achieving up to 4% higher test accuracy for Barlow Twins when 0.1% labelled data is available (i.e. 5 examples per CIFAR-10 class), and that performance drops off dramatically for

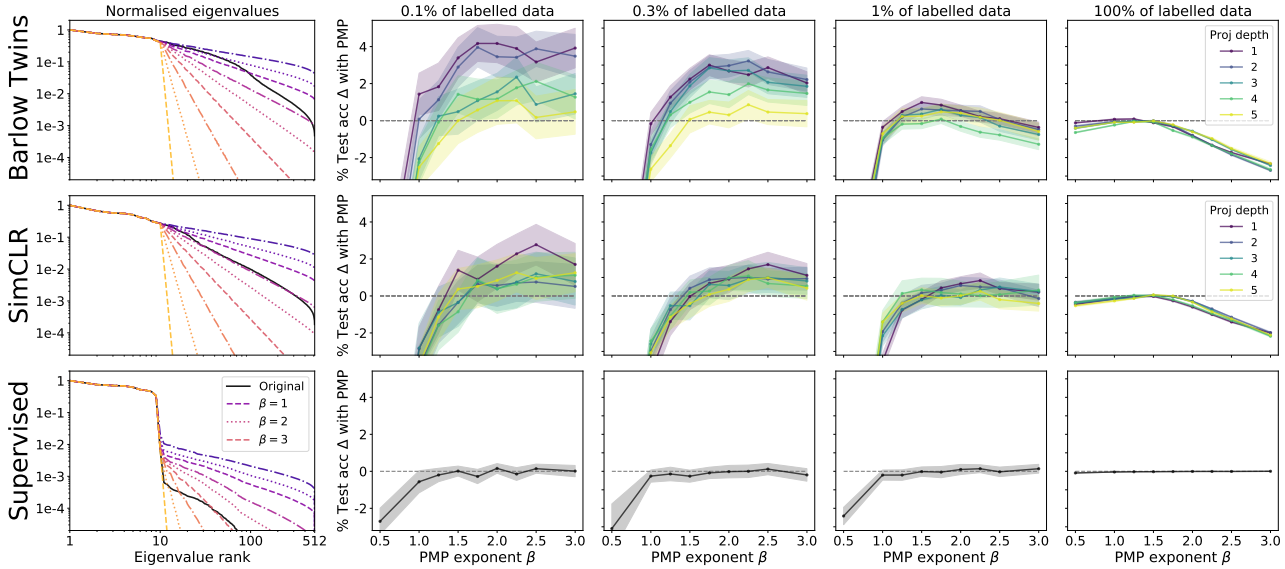

Figure 6. PMP eigenspectra (left) & CIFAR-10 test acc. relative to standard linear probe for varying labelled data sizes across methods.

low β when the features are more whitened. On the other hand, when more labelled data is available, we observe drop-offs in performance using PMP with larger β, although with a well-tuned β, PMP always at least matches the standard linear probe in test-accuracy across all settings. These observations are to be expected from Corollary 4.3: suppressing the useful tail eigenmodes is helpful when not enough labelled data is available to learn them, but can harm performance when there is sufficient labelled data.

We also observe that encoders trained with deeper projection layers benefit less from PMP in small S regimes. This is again suggested by Corollary 3.4, as deeper projection layers already have more collapsed encoder eigenspectra.

Finally, we note that PMP does not seem to be very effective for supervised encoders, which may be because we observe in Figure 6 (bottom left) that supervised training encourages the ResNet-18 encoder to have 10 dominant eigenvalues (corresponding to the 10 classes of CIFAR-10, c.f. Figure 8), and that increasing the size of the smaller eigenvalues via lower β hurts performance, suggesting that the smaller eigenmodes contain unuseful features.

ImageNet-1K For a given pretrained ResNet-50 encoder $h_{\boldsymbol{\theta}}$, we compare PMP to other evaluation schemes on differing amounts of ImageNet-1K labelled data. We compare PMP both to baseline schemes that are also linear in the pretrained encoder $h_{\boldsymbol{\theta}}$: i) standard linear probe (LP), & ii) replacing W_C with a 2-layer linear MLP (MLP) in Eq. (1), as well as non-linear finetuning (NFT), where encoder $h_{\boldsymbol{\theta}}$ is also trainable. We use the same data splits for 1% & 10% as provided by Chen et al. (2020a), and for 0.3%, we sample 3 independent subsets from the 1% split to provide standard deviations. We use a validation split from the accessible training labels to tune hyperparameters for all evaluation schemes, c.f. Appendix C.

In Table 1, we see that PMP improves considerably over other linear evaluations (LP & MLP) over a range of labelled-data sizes & SSL methods: Barlow Twins, SimCLR, & SwAV (Caron et al., 2020). Moreover, we see that PMP consistently outperforms NFT with Barlow Twins & SwAV encoders for smaller values of labelled data, despite keeping $h_{\boldsymbol{\theta}}$ fixed. For example, with 0.3% labelled data & Barlow Twins encoder, PMP obtains 42.3% top-1 accuracy compared to 40.7% for NFT. Likewise, for 1% labelled data & SwAV, PMP beats NFT by over 2.5% in top-1 accuracy. Our top-1 accuracy of 56.2% (obtained via PMP with Barlow Twins pretraining) is to our knowledge the best reported result for an SSL-pretrained ResNet-50 encoder linearly evaluated on 1% ImageNet-1K labels.

Interestingly, standard LP with SwAV can also outperform NFT under label scarcity, where encoder updates may be susceptible to intefering with the useful features acquired during SSL pretraining. However, we also find that NFT outperforms all linear evaluation schemes for SimCLR. Additional experiments are provided in Appendix B including: analysis of the effect of weight decay on PMP in low-labelled data (Figure 10) and evaluation on the ImageNetV2 (Recht et al., 2019) test sets (Table 4).

Transfer Learning We next investigate the ability of PMP to improve transferability of SSL features to new datasets. Using ResNet-50 encoders pretrained on ImageNet-1K, we compare linear probing against PMP on a variety of downstream image classification tasks: CIFAR-100 (Krizhevsky, 2009), Stanford Cars (Krause et al., 2013) and Oxford 102 Flowers (Nilsback & Zisserman, 2008)

Table 1. ImageNet-1K validation accuracy (%) of PMP against standard SSL evaluation schemes, across pretrained checkpoints, with low labelled-data (0.3%, 1%, or 10% labels). Supervised results are from Zhai et al. (2019). Top-1 accuracies with LP on 100% labelled-data are in brackets. 1st & 2nd best results per SSL method & labelled-data level are **bold** & underlined respectively.

Method		Top-1			Top-5		
Pretrain	Eval	0.3%	1%	10%	0.3%	1%	10%
SimCLR (69.3)	LP	$34.2_{\pm.2}$	48.1	61.0	$57.2_{\pm.3}$	73.8	84.3
	MLP	$31.8_{\pm.4}$	45.2	61.5	$54.1_{\pm.3}$	71.1	85.0
	PMP	$\underline{35.9}_{\pm.2}$	$\underline{50.9}$	$\underline{62.5}$	$\underline{57.9}_{\pm.2}$	$\underline{76.6}$	$\underline{85.2}$
	NFT	$\mathbf{39.8}_{\pm.2}$	**52.5**	**67.5**	$\mathbf{65.5}_{\pm.2}$	**78.9**	**88.7**
SwAV (74.7)	LP	$\underline{36.5}_{\pm.3}$	$\underline{53.8}$	68.2	$\underline{61.8}_{\pm.1}$	78.8	88.8
	MLP	$34.6_{\pm.4}$	52.0	67.5	$59.3_{\pm.2}$	77.2	88.6
	PMP	$\mathbf{39.3}_{\pm.3}$	**55.9**	$\underline{68.5}$	$\mathbf{64.1}_{\pm.2}$	**79.7**	$\underline{89.1}$
	NFT	$32.4_{\pm.3}$	53.6	**70.8**	$57.8_{\pm.2}$	$\underline{79.1}$	**90.5**
Barlow (73.5)	LP	$39.9_{\pm.1}$	55.0	63.2	$63.8_{\pm.2}$	79.0	83.4
	MLP	$37.6_{\pm.1}$	53.0	66.3	$61.5_{\pm.1}$	76.9	87.2
	PMP	$\mathbf{42.3}_{\pm.1}$	**56.2**	$\underline{67.3}$	$\mathbf{65.8}_{\pm.1}$	**79.7**	$\underline{88.6}$
	NFT	$\underline{40.7}_{\pm.1}$	$\underline{55.3}$	**70.0**	$\mathbf{65.8}_{\pm.1}$	$\underline{79.6}$	**89.9**
Supervised		–	25.4	56.4	–	48.4	80.4

in Table 2. Across transfer datasets and pretrained SSL encoders, we observe that our PostMan-Pat (PMP) outperforms linear probe (LP) evaluation. We use 100% labelled training data from the transfer dataset in Table 1, demonstrating that while one motivation for PMP was for low-labelled data (Section 4.1), PMP can still outperform LP on large-labelled data settings.

Table 2. **Transfer Learning**: Comparison of top-1 test accuracies (%) for PMP and LP across SSL methods and transfer datasets.

Method		Transfer Dataset		
Pretrain	Eval	C-100	Cars	Flowers
SimCLR	LP	65.26	46.88	84.65
	PMP	$\underline{66.13}$	$\underline{47.83}$	$\underline{85.88}$
Barlow	LP	74.19	69.36	92.29
	PMP	$\underline{75.10}$	$\underline{69.67}$	$\underline{92.54}$
SwAV	LP	75.24	63.39	90.47
	PMP	$\underline{76.10}$	$\underline{64.46}$	$\underline{92.00}$

Different Architectures To assess whether PMP is able to improve label efficiency across different encoder architectures, Table 3 shows compares linear probing and PMP for a ViT-B/16 vision transformer (Dosovitskiy et al., 2020) pretrained using MoCo-v3 (Chen et al., 2021) on ImageNet-1K. We see that PMP consistently improves against LP for ViT-B/16 with lower labelled settings, e.g. 55.2% vs. 54.0% Top-1 on 0.3% labels, and 65.1% vs. 64.5% on 1% labels, although the improvement is more modest on 10% labels. Understanding better how the encoder architecture impacts SSL features (beyond Corollary 3.4), and hence PMP's effectiveness, is interesting future work.

Table 3. **Different Architecture:** PMP and LP ImageNet test accuracies on 1% and 10% labels using ViT-B/16 with MoCo-v3.

	Top-1			Top-5		
Eval	0.3%	1%	10%	0.3%	1%	10%
LP	$54.0_{\pm.3}$	64.5	72.3	$78.0_{\pm.3}$	86.6	91.2
PMP	$\underline{55.2}_{\pm.2}$	$\underline{65.1}$	$\underline{72.4}$	$\underline{78.7}_{\pm.2}$	$\underline{86.8}$	91.2

7. Summary & Discussion

Inspired by work on feature whitening self-supervised learning (SSL) to avoid collapse, we explored the gap between whitening & collapse in SSL. We identified power law behaviour in feature eigenspectra decay as a possible way to bridge between these extremes, and studied the design choices in SSL that affect the rate at which feature eigenvalues decay. We demonstrated theoretically & empirically that weaker regularisation in Barlow Twins & deeper projector layers lead to more collapsed encoders. Moreover, we found empirically that generalisation performance in SSL is non-monotonic in the degree of feature whitening, & highlighted the significance of considering feature eigenspectrum decay in label scarce settings. Finally, we used our insights to motivate a novel post-processing method: PostMan-Pat (PMP) that efficiently enforces a power law in encoder eigenvalues, & demonstrated the ability of PMP to outperform other linear schemes (consistently) & non-linear finetuning (at times) across low labelled-data settings. We hope that the improved label efficiency of PMP can be applied to practical settings of label scarcity. More generally, we hope that our work leads to further progress in SSL by highlighting both the spectrum that exists between collapsed & whitened features, and that where one lies along this spectrum is significant for generalisation performance & label efficiency.

We point out that although we provide empirical evidence from practical settings to corroborate our theoretical results, our theory has some non-standard assumptions to ease analytical exposition, such as linear projector MLPs, much like related theoretical work in SSL (Tian et al., 2021; Wang et al., 2021; Jing et al., 2021). Moreover, we have not studied the actual features acquired during SSL pretraining, e.g. the impact of transformation choice, instead showing that one can improve generalisation accuracy simply by rescaling pre-defined features. Finally, compared to standard linear probes, PMP introduces a new hyperparameter β which needs to be tuned, although it is worth noting that PMP has far fewer significant hyperparameters than non-linear finetuning. For future work, it would be interesting to design an SSL method that directly factors in the rate of feature eigenvalue decay into the pretraining regime, & also to study tuning schemes for PMP hyperparameters.

References

Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

Bordelon, B., Canatar, A., and Pehlevan, C. Spectrum dependent learning curves in kernel regression and wide neural networks. In *International Conference on Machine Learning*, pp. 1024–1034. PMLR, 2020.

Bordes, F., Balestriero, R., and Vincent, P. High fidelity visualization of what your self-supervised representation knows about. *arXiv preprint arXiv:2112.09164*, 2021.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 132–149, 2018.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33:22243–22255, 2020b.

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

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9640–9649, 2021.

Cui, H., Loureiro, B., Krzakala, F., and Zdeborová, L. Generalization error rates in kernel regression: The crossover from the noiseless to noisy regime. *arXiv preprint arXiv:2105.15004*, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

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

Field, D. J. Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Am. A*, 4(12):2379–2394, Dec 1987.

Foster, A., Pukdee, R., and Rainforth, T. Improving transformation invariance in contrastive representation learning. In *International Conference on Learning Representations*, 2021.

Goyal, P., Duval, Q., Reizenstein, J., Leavitt, M., Xu, M., Lefaudeux, B., Singh, M., Reis, V., Caron, M., Bojanowski, P., Joulin, A., and Misra, I. Vissl, 2021.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Hayou, S., Doucet, A., and Rousseau, J. On the impact of the activation function on deep neural networks training. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2672–2680. PMLR, 09–15 Jun 2019.

Hayou, S., Clerico, E., He, B., Deligiannidis, G., Doucet, A., and Rousseau, J. Stable resnet. In *International Conference on Artificial Intelligence and Statistics*, pp. 1324–1332. PMLR, 2021.

He, B., Lakshminarayanan, B., and Teh, Y. W. Bayesian deep ensembles via the neural tangent kernel. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1010–1022. Curran Associates, Inc., 2020a.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020b.

Hua, T., Wang, W., Xue, Z., Ren, S., Wang, Y., and Zhao, H. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9598–9608, 2021.

Izmailov, P., Nicholson, P., Lotfi, S., and Wilson, A. G. Dangers of bayesian model averaging under covariate shift. *arXiv preprint arXiv:2106.11905*, 2021.

Ji, Z. and Telgarsky, M. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2018.

Jin, H., Banerjee, P. K., and Montúfar, G. Learning curves for gaussian process regression with power-law priors and targets. *arXiv preprint arXiv:2110.12231*, 2021.

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

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32: 8572–8583, 2019.

Lee, J., Schoenholz, S., Pennington, J., Adlam, B., Xiao, L., Novak, R., and Sohl-Dickstein, J. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems*, 33, 2020.

Li, Y., Pogodin, R., Sutherland, D. J., and Gretton, A. Self-supervised learning with kernel dependence maximization. *arXiv preprint arXiv:2106.08320*, 2021.

Maennel, H., Alabdulmohsin, I. M., Tolstikhin, I. O., Baldock, R., Bousquet, O., Gelly, S., and Keysers, D. What do neural networks learn when trained with random labels? In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19693–19704. Curran Associates, Inc., 2020.

Martens, J., Ballard, A., Desjardins, G., Swirszcz, G., Dalibard, V., Sohl-Dickstein, J., and Schoenholz, S. S. Rapid training of deep neural networks without skip connections or normalization layers using deep kernel shaping. *arXiv preprint arXiv:2110.01765*, 2021.

Mercer, J. Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations. *Philosophical Transactions of the Royal Society of London Series A*, 209:415–446, January 1909. doi: 10.1098/rsta.1909.0016.

Nassar, J., Sokol, P., Chung, S., Harris, K. D., and Park, I. M. On 1/n neural representation and robustness. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6211–6222. Curran Associates, Inc., 2020.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Papyan, V., Han, X. Y., and Donoho, D. L. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020. ISSN 0027-8424. doi: 10.1073/pnas.2015509117. URL https://www.pnas.org/content/117/40/24652.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019.

Ruderman, D. and Bialek, W. Statistics of natural images: Scaling in the woods. In Cowan, J., Tesauro, G., and Alspector, J. (eds.), *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, 1994.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. Deep information propagation. *arXiv preprint arXiv:1611.01232*, 2016.

Shankar, V., Fang, A., Guo, W., Fridovich-Keil, S., Ragan-Kelley, J., Schmidt, L., and Recht, B. Neural kernels without tangents. In *International Conference on Machine Learning*, pp. 8614–8623. PMLR, 2020.

Sollich, P. Learning curves for gaussian processes. In Kearns, M., Solla, S., and Cohn, D. (eds.), *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1999.

Sollich, P. Gaussian process regression with mismatched models. In Dietterich, T., Becker, S., and Ghahramani, Z. (eds.), *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2002.

Sollich, P. and Halees, A. Learning curves for gaussian process regression: Approximations and bounds. *Neural computation*, 14(6):1393–1428, 2002.

Stringer, C., Pachitariu, M., Steinmetz, N., Carandini, M., and Harris, K. D. High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765): 361–365, 2019.

Tian, Y., Chen, X., and Ganguli, S. Understanding self-supervised learning dynamics without contrastive pairs. *arXiv preprint arXiv:2102.06810*, 2021.

Wang, X., Chen, X., Du, S. S., and Tian, Y. Towards demystifying representation learning with non-contrastive self-supervision. *arXiv preprint arXiv:2110.04947*, 2021.

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

Zhai, X., Oliver, A., Kolesnikov, A., and Beyer, L. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1476–1485, 2019.

Zhang, C., Zhang, K., Zhang, C., Pham, T. X., Yoo, C. D., and Kweon, I. S. How does simsiam avoid collapse without negative samples? a unified understanding with self-supervised contrastive learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=bwq6O4Cwdl.

A. Proofs & Additional Results

Throughout, we use o, O, Θ to denote standard mathematical notation for order size.

A.1. Proposition 3.1

We start by proving Proposition 3.1.

Proposition 3.1. *Suppose an NN encoder+projection trained via Barlow Twins (Zbontar et al., 2021) achieves (i) training loss $\mathcal{L}_{BT} = \epsilon$, & ii) $\exists a \leq b$ positive constants such that $a \leq \Sigma^z_{ii} \leq b$, $\forall i \in [d_p]$. Then, if $\epsilon \leq \frac{a^2\rho}{b^2}$, the projector has a whitened eigenspectrum.*

Proof. Because $\mathcal{L}_{\text{BT}} = \epsilon$, we see that the off-diagonal contribution to the loss satisfies:

$$\sum_{i=1}^{d_p} \sum_{j \neq i} C_{i,j}^2 \leq \frac{\epsilon}{\rho}.$$

From Eq. (3), we see that $C_{ij} = \frac{\Sigma^z_{i,j}}{\sqrt{\Sigma^z_{i,i}\Sigma^z_{j,j}}}$ up to transformations (see remark below), and from assumption ii), we deduce that:

$$\sum_{i=1}^{d_p} \sum_{j \neq i} (\Sigma^z_{i,j})^2 \leq \frac{b^2\epsilon}{\rho}.$$

Thus, if $\text{diag}(\Sigma^z)$ denotes the diagonal version of Σ^z , then we see that

$$\|\text{diag}(\Sigma^z) - \Sigma^z\|_F^2 \leq \frac{b^2\epsilon}{\rho}.$$

Recall that the squared Frobenius norm of a matrix is also the sum of its squared eigenvalues. So if $\frac{b^2\epsilon}{\rho} < a^2$, then we have a whitened projection eigenspectrum of Σ^z, with new constants $a' = \sqrt{a^2 - \frac{b^2\epsilon}{\rho}}$ and $b' = \sqrt{b^2 + \frac{b^2\epsilon}{\rho}}$ satisfying Def. 2.1.

□

Remark A.1. Though it is not strictly true that $C_{ij} = \frac{\Sigma^z_{i,j}}{\sqrt{\Sigma^z_{i,i}\Sigma^z_{j,j}}}$ due to the input transformations in C_{ij}, we note that test-time feature-averaging across transformations has been shown to be an effective method (at the expense of extra forward passes) to improve generalisation of contrastive SSL (Foster et al., 2021).

In this case, we can replace features $z(\boldsymbol{x}) \in \mathbb{R}^{d_p}$ with augmented features $Z(\boldsymbol{x}) \in \mathbb{R}^{d_p \times K}$, where $Z(\boldsymbol{x})_{i,k} = z_i(T_k(\boldsymbol{x}))$, corresponding to K independently sampled transformations $\{T_k\}_{k=1}^K \overset{\text{i.i.d.}}{\sim} \mathcal{T}$.

Then, the natural covariance definition $\Sigma^Z_{i,j} = \frac{1}{NK}\sum_{n=1}^N \left(Z(\boldsymbol{x}_n)Z(\boldsymbol{x}_n)^T\right)_{i,j}$ does satisfy $C_{ij} = \frac{\Sigma^Z_{i,j}}{\sqrt{\Sigma^Z_{i,i}\Sigma^Z_{j,j}}}$ in the $K \to \infty$ limit.

A.2. Proposition 3.3

We next prove Proposition 3.3, restated below:

Proposition 3.3. *Suppose we have an L-layer linear MLP, $f(\boldsymbol{x}) = \prod_{l=1}^L W_l \cdot \boldsymbol{x} \in \mathbb{R}^{d_L}$, trained to convergence using gradient flow & no bias terms on some loss $\mathcal{L}(f(\boldsymbol{X}))$ with weight decay $\eta > 0$. Assume further that the first layer matrix $W_1 \in \mathbb{R}^{d_1 \times d}$ & input covariance matrix $\Sigma^{\boldsymbol{x}} \in \mathbb{R}^{d \times d}$ are aligned as in Def. 3.2. Then:*

1. *Adjacent layers' matrices W_l & W_{l-1} become aligned during training for $1 < l \leq L$ so that principal components can be grouped together across layers. If $\lambda_{l,j}$ denotes the j^{th} eigenvalue of the empirical covariance of features at layer l, then:*

2. *For any uncollapsed output eigenvalue j with $\lambda_{L,j}>0$ & any two layers $0\leq k<l$ (where $k=0$ denotes the input layer), we have:*

$$\lambda_{l,j}=(\lambda_{k,j})^{\frac{L-l}{L-k}}(\lambda_{L,j})^{\frac{l-k}{L-k}},$$

i.e. $\lambda_{l,j}$ is a weighted geometric mean between $\lambda_{k,j}$ & $\lambda_{L,j}$, with weighting specified by closeness to k or L.

Proof. If our linear MLP is $f(\boldsymbol{x})=\prod_{l=1}^{L}W_l\cdot\boldsymbol{x}$ with weight matrices $W_l\in\mathbb{R}^{d_l\times d_{l-1}}$, then gradient flow on loss $\mathcal{L}$ with weight decay $\eta>0$ yields dynamics (c.f. (Ji & Telgarsky, 2018)):

$$\dot{W}_l=-W_{l+1}^T\cdots W_L^T\chi^T W_1^T\cdots W_{l-1}^T-\eta W_l, \tag{6}$$

$\forall l\geq 1$, where:

$$\chi=\boldsymbol{X}^T\frac{\partial\mathcal{L}(f)}{\partial f}\bigg|_{f=f(\boldsymbol{X})}\in\mathbb{R}^{d\times d_L}. \tag{7}$$

We see from Eq. (6) that:

$$W_l^T\dot{W}_l+\eta W_l^T W_l=\dot{W}_{l-1}W_{l-1}^T+\eta W_{l-1}W_{l-1}^T, \tag{8}$$

and taking the transpose of Eq. (8) yields:

$$\dot{W}_l^T W_l+\eta W_l^T W_l=W_{l-1}\dot{W}_{l-1}^T+\eta W_{l-1}W_{l-1}^T, \tag{9}$$

Taking the sum of Eqs. (8) and (9), and integrating both sides with respect to time gives us:

$$W_l^T W_l=W_{l-1}W_{l-1}^T+C_l e^{-2\eta t}, \tag{10}$$

for some constant matrix C_l. Thus, we see that $\lim_{t\to\infty}W_l^T W_l-W_{l-1}W_{l-1}^T=0$. If we let $U_l D_l V_l^T$ denote the Singular Value Decomposition (SVD) of W_l in this limit, then (noting that SVDs are unique only up to permutations of the principal components, so we are free to choose a permutation that ensures alignment) we see that:

- $V_l^T U_{l-1}=I_{d_{l-1}},\ \ \forall l\geq 2.$
- $D_L=\cdots=D_l=D_{l-1}=\cdots=D_1.$

This concludes the proof for 1).

For 2), recall we assumed that $V_1^T U_{\boldsymbol{x}}=I$, where the empirical input covariance has SVD $\Sigma^{\boldsymbol{x}}=U_{\boldsymbol{x}}D_{\boldsymbol{x}}^2U_{\boldsymbol{x}}^T$. For simplicity, we can assume that the input distribution is centred $\frac{1}{N}\sum_n\boldsymbol{x}_n=0$ without loss of generality due to linearity, which ensures that the empirical output distribution is too: $\frac{1}{N}\sum_n f(\boldsymbol{x}_n)=0$.

Thus by construction, the empirical input covariance is $\Sigma^{\boldsymbol{x}}=\frac{1}{N}\sum_n\boldsymbol{x}_n\boldsymbol{x}_n^T$, so we can put everything together and calculate the output covariance to be:

$$\Sigma^L=\frac{1}{N}\sum_n f(\boldsymbol{x}_n)f(\boldsymbol{x}_n)^T\in\mathbb{R}^{d_L\times d_L} \tag{11}$$

$$=W_L\cdots W_1\cdot\Sigma^{\boldsymbol{x}}\cdot W_1^T\cdots W_L^T \tag{12}$$

$$=U_L\bigg(\prod_{l=1}^{L}D_l^2\bigg)D_{\boldsymbol{x}}^2U_L^T \tag{13}$$

$$=U_L D_1^{2L}D_{\boldsymbol{x}}^2U_L^T \tag{14}$$

Let us denote $\gamma_j=(D_1)_{j,j}^2$ and $\lambda_{0,j}=(D_{\boldsymbol{x}})_{j,j}^2$, then we see from Eq. (14) that $\lambda_{L,j}=\gamma_j^L\lambda_{0,j}$, such that $\gamma_j=(\frac{\lambda_{L,j}}{\lambda_{0,j}})^{\frac{1}{L}}$.

In a similar vein to Eqs. (11) to (14), it is simple to calculate the empirical covariance at layer $l \geq 0$ by

$$\Sigma^l = U_l D_1^{2l} D_{\boldsymbol{x}}^2 U_l^T. \tag{15}$$

We thus conclude that $\lambda_{l,j} = \lambda_{0,j}(\frac{\lambda_{L,j}}{\lambda_{0,j}})^{\frac{l}{L}} = (\lambda_{0,j})^{\frac{L-l}{L}}(\lambda_{L,j})^{\frac{l}{L}} \quad \forall l$, from which the result 2) follows easily, by noting that $\left(\frac{\lambda_{l,j}}{(\lambda_{L,j})^{\frac{l}{L}}}\right)^{\frac{1}{L-l}}$ is constant in l: i.e. one can set $\left(\frac{\lambda_{l,j}}{(\lambda_{L,j})^{\frac{l}{L}}}\right)^{\frac{1}{L-l}} = \left(\frac{\lambda_{k,j}}{(\lambda_{L,j})^{\frac{k}{L}}}\right)^{\frac{1}{L-k}}$ and simplify. □

A.3. Theorem A.2

Finally, we state and prove Theorem A.2. Suppose that we have two kernels k_1 and k_2, with which we want to learn two (potentially different) functions f_1^* and f_2^* respectively using kernel ridge regression. If one kernel has whitened eigenvalues but the other does not, then the whitened kernel has relatively worse generalisation error at a particular value of labelled data $S^* < \infty$, under certain assumptions. Note Theorem A.2 does not require power law assumptions on f_a^*, unlike Proposition 4.2.

The intuition is that all eigenmodes are learnt as $S \to \infty$, but at a large but intermediate value of S^* (depending on ridge parameter σ^2 and the level of imbalance in eigenvalue sizes), only the largest eigenvalues are learnt. As it is harder to learn the smaller eigenvalues (in the sense that we need more data to learn the same size eigenmode μ_i^2 if eigenvalue λ_i is smaller, c.f. Eq. (20)) compared to larger eigenvalues, at this intermediate value of S^* it is possible to have relatively lower error at the small eigenmodes by not learning the small eigenmodes (and only learning the dominant eigenmodes), compared to larger values of S:

Theorem A.2. *For $a \in [2]$, let kernel k_a have sorted (in decreasing order) eigenvalues $\{\lambda_{a,i}\}_{i=1}^{d_a}$ & eigenfunctions $\{\psi_{a,i}\}_{i=1}^{d_a}$. Let f_a^* be such that $\exists \{\mu_{a,i}\}_{i=1}^{d_a}$ satisfying Eq. (5) with $\{\psi_{a,i}\}_i$, i.e. $f_a^*(\boldsymbol{x}) = \sum_{i=1}^{d_a} \mu_{a,i}\psi_{a,i}(\boldsymbol{x})$.*

Suppose $\frac{\mu_{a,i}^2}{\lambda_{a,i}^2} = \Theta(1), \quad \forall a \in [2], i \in [d_a]$, and also that $\sigma^2 = \Theta(1)$ is fixed.

Suppose also $\{\lambda_{2,i}\}_i$ are whitened, but $\{\lambda_{1,i}\}_i$ are not, in that $\exists i \in [d_1]$, and $M > 0$ large satisfying $\frac{\lambda_{1,1}}{\lambda_{1,i}} = M > 0$ and $\lambda_{1,1} = \Theta(1)$.

If we try to learn f_a^* with kernel k_a via ridge regression with ridge parameter σ^2, and let $\hat{\mathcal{E}}_S^a$ denote the associated generalisation error.

Then:

1. *$\frac{\hat{\mathcal{E}}_S^1}{\hat{\mathcal{E}}_S^2} \xrightarrow{S\to\infty} C$, where $C > 0$ satisfies $\sum_{i=1}^{d_1} \frac{\mu_{1,i}^2}{\lambda_{1,i}^2} = C \sum_{i=1}^{d_2} \frac{\mu_{2,i}^2}{\lambda_{2,i}^2}$.*
2. *Moreover, if M is large enough, then $\exists S^* < \infty$ depending on σ^2 and λ_i s.t. $\frac{\hat{\mathcal{E}}_{S^*}^1}{\hat{\mathcal{E}}_{S^*}^2} < C$.*

Proof. For $a = 1, 2$, let us define

$$w_{a,i} = \frac{\mu_{a,i}}{\sqrt{\lambda_{a,i}}},$$

such that $f_a^*(\boldsymbol{x}) = \sum_{i=1}^{d_a} w_{a,i}\sqrt{\lambda_{a,i}}\psi_{a,i}(\boldsymbol{x})$ and let $\hat{\mathcal{E}}_{i,S}^a$ denote the generalisation error associated to mode i for learning function f_a^*, so that by orthogonality, we have:

$$\hat{\mathcal{E}}_S^a = \sum_i \hat{\mathcal{E}}_{i,S}^a. \tag{16}$$

Then, from Bordelon et al. (2020) (Proposition 3), we have the following approximation:

$$\hat{\mathcal{E}}_{i,S}^a = \frac{w_{a,i}^2}{\lambda_{a,i}}\left(\frac{1}{\lambda_{a,i}} + \frac{S}{\sigma^2 + t_a(S)}\right)^{-2}\left(1 - \frac{S\gamma_a(S)}{(\sigma^2 + t_a(S))^2}\right)^{-1} \tag{17}$$

where $t_a(S)$ is the solution to the implicit equation

$$t_a(S) = \sum_{i=1}^{d_a} \left(\frac{1}{\lambda_{a,i}} + \frac{S}{\sigma^2 + t_a(S)} \right)^{-1}, \tag{18}$$

and $\gamma_a(S)$ is defined by

$$\gamma_a(S) \triangleq \sum_{i=1}^{d_a} \left(\frac{1}{\lambda_{a,i}} + \frac{S}{\sigma^2 + t_a(S)} \right)^{-2}. \tag{19}$$

Note that, $0 \leq t_a(S) \leq \sum_{i=1}^{d_a} \lambda_{a,i} < \infty$, and likewise $0 \leq \gamma_a(S) \leq \sum_{i=1}^{d_a} \lambda_{a,i}^2 < \infty$. Moreover, we have $\gamma_a(S) \leq \Theta(S^{-2})$. Also note that $t_a(S) = o(1)$ as $S \to \infty$.

Then, as $S \to \infty$, we have from Eq. (17):

$$\hat{\mathcal{E}}^a_{i,S} = \frac{\sigma^4 w_{a,i}^2}{S^2 \lambda_{a,i}} (1 + o(1)). \tag{20}$$

So from Eq. (16), we see that

$$\frac{\hat{\mathcal{E}}^1_S}{\hat{\mathcal{E}}^2_S} \xrightarrow{S \to \infty} C \tag{21}$$

where C satisfies

$$\sum_{i=1}^{d_1} \frac{w_{1,i}^2}{\lambda_{1,i}} = C \sum_{i=1}^{d_2} \frac{w_{2,i}^2}{\lambda_{2,i}}. \tag{22}$$

This concludes the proof for 1).

Now we are given that $M\lambda_{1,i} = \lambda_{1,1} = \Theta(1)$ for large M.

Suppose that we have S^* satisfying:

$$1 \ll \frac{S^*}{\sigma^2 + t_a(S^*)} \ll M, \quad \forall a \in [2]. \tag{23}$$

We know that such an S^* exists as $\sigma^2 \leq \sigma^2 + t_a(S) \leq \sigma^2 + \sum_{i=1}^{d_a} \lambda_{a,i} < \infty, \forall S, a \in [2]$.

Then, we have:

$$\frac{\hat{\mathcal{E}}^1_{i,S^*}}{\hat{\mathcal{E}}^1_{1,S^*}} = \frac{w_{1,i}^2 \lambda_{1,1}}{w_{1,1}^2 \lambda_{1,i}} \left(\frac{1}{\lambda_{1,i}} + \frac{S^*}{\sigma^2 + t_1(S^*)} \right)^{-2} \left(\frac{1}{\lambda_{1,1}} + \frac{S^*}{\sigma^2 + t_1(S^*)} \right)^{2} \tag{24}$$

$$= \frac{w_{1,i}^2 \lambda_{1,1}}{w_{1,1}^2 \lambda_{1,i}} \left(\frac{\lambda_{1,i} S^*}{\sigma^2 + t_1(S^*)} \right)^{2} \left(1 + O(\frac{1}{S^*}) + O(\frac{S^*}{M}) \right), \tag{25}$$

and by construction, we have

$$\frac{\lambda_{1,i} S^*}{\sigma^2 + t_1(S^*)} \ll 1,$$

so we see that:

$$\frac{\hat{\mathcal{E}}^1_{i,S^*}}{\hat{\mathcal{E}}^1_{1,S^*}} < \frac{w^2_{1,i}\lambda_{1,1}}{w^2_{1,1}\lambda_{1,i}} \tag{26}$$

$$= \lim_{S\to\infty} \frac{\hat{\mathcal{E}}^1_{i,S}}{\hat{\mathcal{E}}^1_{1,S}} \quad \text{from Eq. (20)} \tag{27}$$

up to $\left(1 + O(\frac{1}{S^*}) + O(\frac{S^*}{M})\right)$ multiplicative error, as $\frac{w^2_{1,i}}{\lambda_{1,i}} = \frac{\mu^2_{1,i}}{\lambda^2_{1,i}} = \Theta(1) > 0$ by assumption.

Likewise, for $j \notin \{i, 1\}$, we know $\lambda_{1,j} \leq \lambda_{1,1}$ (as eigenvalues are sorted so $\lambda_{1,1}$ is the largest eigenvalue), so that from Eq. (24) (replacing i with j):

$$\frac{\hat{\mathcal{E}}^1_{j,S^*}}{\hat{\mathcal{E}}^1_{1,S^*}} \leq \frac{w^2_{1,j}\lambda_{1,1}}{w^2_{1,1}\lambda_{1,j}} = \lim_{S\to\infty} \frac{\hat{\mathcal{E}}^1_{j,S}}{\hat{\mathcal{E}}^1_{1,S}}. \tag{28}$$

On the other hand, for k_2, as $(\lambda_{2,i})_i$ are whitened, i.e. $\lambda_{2,i} = \Theta(1)$, $\forall i$, we have:

$$\frac{\hat{\mathcal{E}}^2_{i,S^*}}{\hat{\mathcal{E}}^2_{1,S^*}} = \lim_{S\to\infty} \frac{\hat{\mathcal{E}}^2_{i,S}}{\hat{\mathcal{E}}^2_{1,S}} \tag{29}$$

up to $\left(1 + O(\frac{1}{S^*})\right)$ multiplicative error.

Finally, we have the following ratio for the dominant eigenmode errors between k_1 and k_2:

$$\frac{\hat{\mathcal{E}}^1_{1,S^*}}{\hat{\mathcal{E}}^2_{1,S^*}} = \frac{w^2_{1,1}\lambda_{2,1}}{w^2_{2,1}\lambda_{1,1}} \left(\frac{1}{\lambda_{1,1}} + \frac{S^*}{\sigma^2 + t_1(S^*)}\right)^{-2} \left(\frac{1}{\lambda_{2,1}} + \frac{S^*}{\sigma^2 + t_2(S^*)}\right)^{2} \times \tag{30}$$

$$\left(1 - \frac{S^*\gamma_1(S^*)}{(\sigma^2 + t_1(S^*))^2}\right)^{-1} \left(1 - \frac{S^*\gamma_2(S^*)}{(\sigma^2 + t_2(S^*))^2}\right) \tag{31}$$

$$\tag{32}$$

but note that by construction in Eq. (23), S^* satisfies $t_1(S^*), t_2(S^*) = o(\sigma^2)$ as we recall $\sigma^2 = \Theta(1)$. Moreover, as both $\lambda_{1,1}$ and $\lambda_{2,1}$ are $\Theta(1)$, we have $\big($up to $(1 + O(\frac{1}{S^*}) + O(\frac{t_1(S^*)+t_2(S^*)}{\sigma^2}))$ multiplicative error$\big)$:

$$\frac{\hat{\mathcal{E}}^1_{1,S^*}}{\hat{\mathcal{E}}^2_{1,S^*}} = \frac{w^2_{1,1}\lambda_{2,1}}{w^2_{2,1}\lambda_{1,1}} \tag{33}$$

$$= \lim_{S\to\infty} \frac{\hat{\mathcal{E}}^1_{1,S}}{\hat{\mathcal{E}}^2_{1,S}}. \tag{34}$$

Putting this all together, for large enough M & S^* satisfying Eq. (23) (such that all $(1 + o(1))$ multiplicative errors may be

ignored):

$$\frac{\hat{\mathcal{E}}^1_{S^*}}{\hat{\mathcal{E}}^2_{S^*}} = \frac{\sum_{m=1}^{d_1} \hat{\mathcal{E}}^1_{m,S^*}}{\sum_{j=1}^{d_2} \hat{\mathcal{E}}^2_{j,S^*}} \tag{35}$$

$$= \frac{\hat{\mathcal{E}}^1_{1,S^*}}{\hat{\mathcal{E}}^2_{1,S^*}} \frac{\sum_{m=1}^{d_1} \frac{\hat{\mathcal{E}}^1_{m,S^*}}{\hat{\mathcal{E}}^1_{1,S^*}}}{\sum_{j=1}^{d_2} \frac{\hat{\mathcal{E}}^2_{j,S^*}}{\hat{\mathcal{E}}^2_{1,S^*}}} \tag{36}$$

$$< \lim_{S\to\infty} \frac{\hat{\mathcal{E}}^1_{1,S}}{\hat{\mathcal{E}}^2_{1,S}} \frac{\sum_{m=1}^{d_1} \frac{\hat{\mathcal{E}}^1_{m,S}}{\hat{\mathcal{E}}^1_{1,S}}}{\sum_{j=1}^{d_2} \frac{\hat{\mathcal{E}}^2_{j,S}}{\hat{\mathcal{E}}^2_{1,S}}} \quad \text{by Eqs. (26), (28), (29) and (34)} \tag{37}$$

$$= \lim_{S\to\infty} \frac{\hat{\mathcal{E}}^1_{S}}{\hat{\mathcal{E}}^2_{S}} \tag{38}$$

$$= C \tag{39}$$

as required. □

B. Additional Experiments

STL-10 analysis Figure 7 is akin to Figure 4, but trained with Barlow Twins on STL-10 dataset. Training hyperparameters matched exactly the values in Figure 7, except slightly different data-augmentations (Color Jitter & Gaussian Blur) were used for SSL pretraining, matching the default values of the codebase in Footnote 2. We observe similar trends in Figure 7 to Figure 4 where deeper projections have more collapsed encoder representations, and also test accuracy is not monotonic in the degree of whitening.

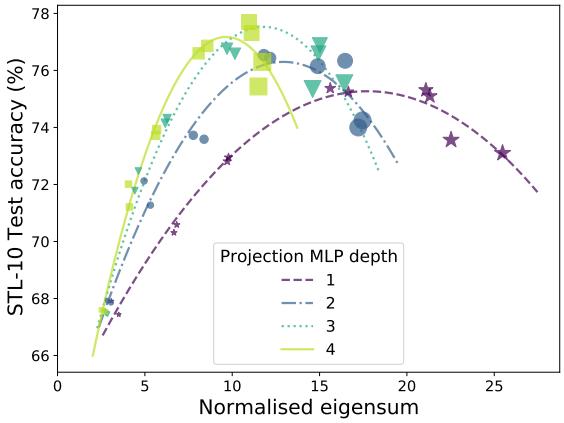

Figure 7. Akin to Figure 4 but using STL-10 as unlabelled dataset for Barlow Twins training as opposed to CIFAR-10.

Gram matrices across methods In Figure 8 we plot feature Gram correlation matrices, with $(i,j)^{\text{th}}$ entry

$$\frac{\langle h_{\boldsymbol{\theta}}(\boldsymbol{x}_i), h_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle}{\|h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\|_2 \|h_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\|_2}$$

over 1000 CIFAR-10 test points, for the 3 pretrained ResNet-18 (either with SimCLR, Barlow Twins, or Supervised) displayed in Figure 6. We see that the NN trained with supervision has feature Gram matrix that is much closer qualitatively to the classwise Gram matrix in $\mathbb{R}^{1000\times 1000}$ (which takes $(i,j)^{\text{th}}$ value 1 if input i and input j are from the same class, and 0

else). As the classwise Gram matrix has exactly C non-zero eigenvalues for C classes, this provides evidence that the feature collapse in Figure 6 (bottom left) is due to the fact that the 10 dominant eigenvalues correspond to the 10 different classes in CIFAR-10. This is consistent with the recently observed Neural Collapse phenomenon (Papyan et al., 2020). On the other hand, without training labels, SSL methods do not have as obvious a correspondence between dominant eigenvalues and classes.

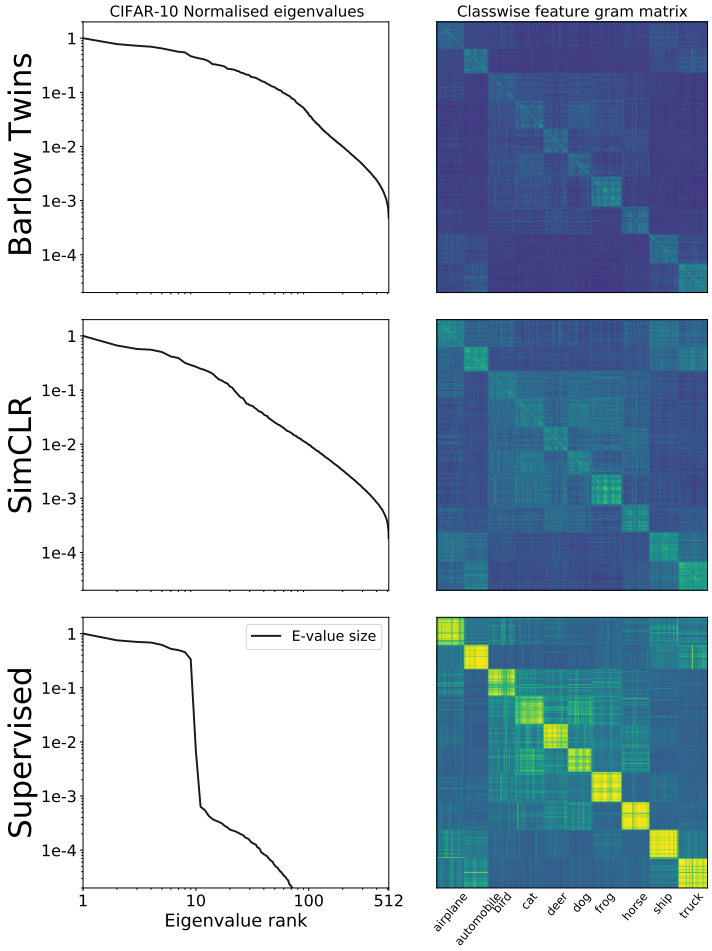

Figure 8. Feature eigenspectra and Gram matrices corresponding to Figure 6, over 1000 CIFAR-10 test examples.

ImageNet-1K with large labelled-data. In Figure 9 (right), we plot the performance of PMP with Barlow Twins pretrained ResNet-50 encoder when given access to all 1.2 million training labels for evaluation. We see that PMP is able to match the performance of standard linear probe of 73.5% at values of $\beta = 0.8$. This is unsurprising given Figure 9 (left), which shows that the encoder eigenspectra (after rank 10) already approximately decays with exponent 0.9. Weight decay 0.0001 is used. It would be interesting to see if one can improve SSL in large labelled data regimes with more optimal tuning of e.g. weight decay, but for this it would also be desirable to first design more efficient methods of PMP hyperparameter tuning.

Interactions between weight decay and β for low labelled data. In Figure 10, we plot the accuracy of PMP with SimCLR, SwAV and Barlow Twins pretrained ResNet-50 encoders with 1% training labels, for different values of weight decay. For smaller values of weight decay, we see that larger β, corresponding to more collapsed features, yield higher top-1 accuracy. This is consistent with the findings of Corollary 4.3, Theorem A.2, and Figure 6, where suppressing the smaller eigenvalues is useful in low labelled data regimes. However, we also find that for larger values of weight decay, this trend is reversed, in that the best values of β are small, hence more whitened eigenspectra perform better. Indeed, the

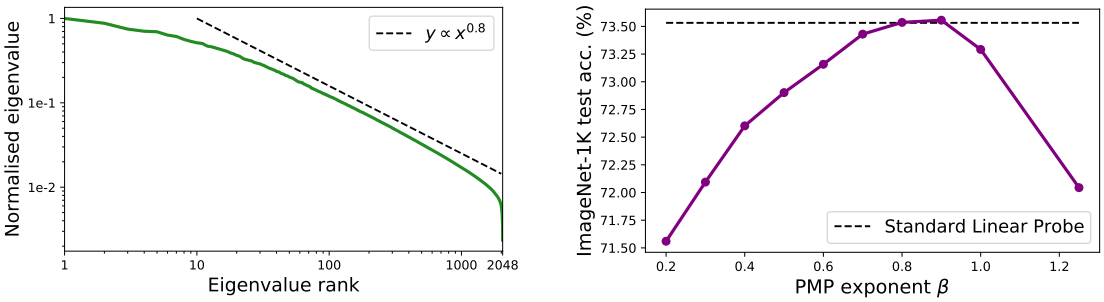

Figure 9. Barlow Twins eigenspectra plot (left) & ImageNet-1k validation accuracy using *all* 1.2 million training labels for PMP as a function of β (right). We see that the best values of β match standard linear probe at the value of $\beta = 0.8$ which matches the rate of decay of the original encoder. The encoder was pretrained with Barlow Twins and taken from the official implementation of Zbontar et al. (2021). Weight decay 0.0001 was used at evaluation time.

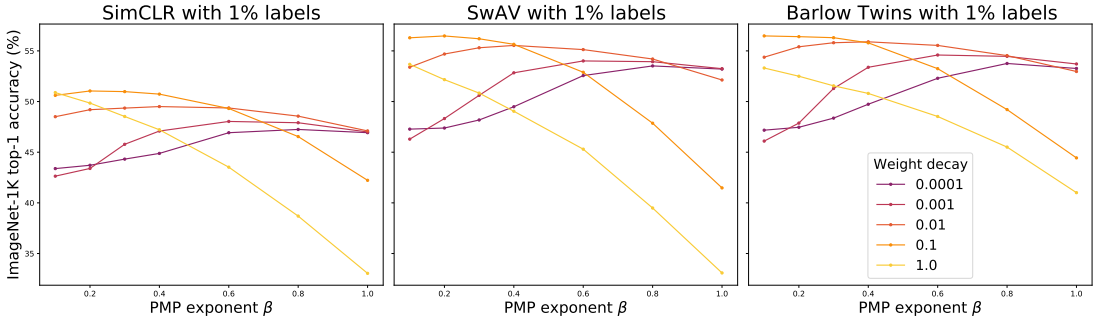

Figure 10. Accuracy in low labelled data setting as a function of PMP power law exponent β for different weight decay settings.

best performing hyperparameter setting chose a relatively large weight decay combined with small β (although we see that too large weight decay also hurts accuracy, particularly for better performing encoders: SwAV and Barlow Twins). This surprising observation is not covered by our theoretical results (which concern fixed weight decay), but does emphasise the importance of where an encoder lies along the gap between collapsed & whitened features, and its decay rate of eigenspectra, in determining its generalisation performance, particularly in low-labelled data settings.

B.1. ImageNetV2 results for PMP

To verify that our results in Table 1 are not overfit to the ImageNet-1K validation set, in Table 4 we provide corresponding results for evaluating PMP in low-labelled settings (from the ImageNet-1K training set) on the ImageNetV2 test datasets. We observe the same trends in Table 4 as in Table 1: PMP always outperforms LP; PMP often outperforms NFT; and LP sometimes outperforms NFT on SwAV.

C. Experimental Details

C.1. Figure 3

Our SimCLR implementation was taken from an open-source codebase[2] & we used default hyperparameters provided like 0.5 temperature for InfoNCE; our Barlow Twins implementation used Alg. 1 of Zbontar et al. (2021).

Both Barlow Twins & SimCLR ResNet-18 encoders used projectors of depth 2, with ReLU & BatchNorm. Barlow Twins used wide projector width 1024 & batch size 256 whereas SimCLR used smaller width 256 & larger batch size 512, which are all standard hyperparameter choices. All networks were trained with SGD loss for 100 epochs with weight decay 0.0004, momentum 0.9 & a cosine annealed learning rate.

[2]https://github.com/facebookresearch/luckmatters

Table 4. **ImageNetV2 test evaluation**: PMP top-1 % accuracy vs. LP & non-linear finetuning (NFT) on the 3 ImageNetv2 test sets: Matched Frequency (MF), Threshold0.7 (T-0.7) & TopImages (Top-I), across low-label settings & SSL methods.

METHOD	LABELS	EVAL	MF	T-0.7	TOP-I
BARLOW	1%	LP	44.18	52.65	58.85
		PMP	**45.60**	**54.11**	**60.03**
		NFT	45.00	53.39	59.06
	10%	LP	51.62	60.40	66.32
		PMP	55.36	64.49	70.65
		NFT	**58.57**	**67.46**	**73.15**
SIMCLR	1%	LP	37.78	45.73	51.58
		PMP	40.89	49.06	54.62
		NFT	**41.76**	**49.80**	**55.63**
	10%	LP	48.62	57.77	64.15
		PMP	49.77	59.26	65.72
		NFT	**54.80**	**64.00**	**69.70**
SWAV	1%	LP	42.46	50.68	57.15
		PMP	**44.24**	**52.44**	**58.63**
		NFT	42.74	50.21	56.70
	10%	LP	55.68	64.76	70.88
		PMP	56.44	65.29	71.54
		NFT	**58.76**	**68.13**	**73.92**

Factors such as learning rate and regularisation strength were chosen to ensure all networks achieved similar test accuracy under linear probe (≈85%). Learning rate was 0.32 for SimCLR & 0.25 for Barlow Twins, with $\rho = 0.01$. All methods used the default data-augmentations on CIFAR-10 as described in Chen et al. (2020a).

C.2. Figure 4

All training details follow Figure 3 above, though we add that the values for ρ used were $\{0.001, 0.003, 0.01, 0.03, 0.05\}$.

C.3. Figure 6

For all PMP experiments, we start eigenvalue decay power law after the tenth largest eigenvalue in PMP, Alg. 1. This is consistent with Nassar et al. (2020), who studied importance of power-law decay in eigenspectra for adversarial robustness, & the findings of Stringer et al. (2019). For the subsets of labelled data (including for 0.3% ImageNet-1K too), we sample uniformly at random from the CIFAR-10 train set, ensuring that all classes have an equal number of examples (so that 0.1% labelled-data corresponds to 5 examples per class). In all cases at evaluation time, for linear probe or PMP, we trained W_C using SGD for 50 epochs with batch size 128, momentum 0.9, weight decay 0.001, learning rate 0.1 and cross-entropy loss.

The supervised NN was trained for 160 epochs using SGD+momentum with learning rate 0.05 and batch size 128. Weight decay for supervised training was set to 0.0003 to ensure the NN also achieved similar CIFAR-10 test accuracy (85%). To that end, standard data augmentation (random crops & flips) was not used to train the supervised NN, and the only preprocessing of images was normalising before training.

C.4. ImageNet-1K evaluation: Table 1

Our ImageNet-1K implementation was based off the official Barlow Twins (Zbontar et al., 2021) implemtation[3], which is also where we obtained the ResNet-50 Barlow Twin checkpoint pretrained on ImageNet-1K. The SimCLR (Chen et al., 2020a) & SwAV (Caron et al., 2020) ResNet-50 checkpoints were obtained from the VISSL library's (Goyal et al., 2021) model zoo. In particular, we selected the SimCLR checkpoint that was trained for 800 epochs, and the SwAV checkpoint that was trained for 800 epochs with multi-crop setting: $2 \times 224 + 6 \times 96$. These selections were based on the best top-1 accuracy performing checkpoints (under linear probe).

[3]https://github.com/facebookresearch/barlowtwins

The top-1 accuracies under linear probe in Table 1 have slight discrepancies to those reported in VISSL ($< 0.4\%$ difference), possibly reflecting slight differences in linear evaluation training schemes e.g. we use weight decay 0.0001 & normalise each of the 2048 features to have zero mean and unit variance across the unlabelled ImageNet-1K dataset. Indeed, we normalised neurons across all linear evaluation schemes & all labelled-data settings, in order to avoid the setting where non-zero means result in a single eigenvalue that is orders of magnitude larger than others.

In all linear evaluation schemes: standard linear probe (LP); 2-layer linear MLP classifier (MLP); & our PostMan-Pat (PMP), we train the classifier for 100 epochs using SGD & momentum 0.9, with a cosine annealed learning rate starting at 0.1, with weight decay tuned in all cases (along with power law exponent β for PMP). For the linear MLP classifier we used single hidden layer of width 4096.

For non-linear finetuning (NFT), we tuned followed the same training procedure, apart from additionally tuning the number of training epochs (between 20 & 40, which is consistent with Zbontar et al. (2021)), as well as separate encoder & classifier learning rates. In NFT, we did not use weight decay for the encoder, as this would remove the useful features learning in the encoder during pretraining. However, we did tune weight decay for the linear classifier W_C.

Hyperparameter tuning data splits For any given set of labelled data, we split the data into 4:1 splits for the 1% or 10% labelled-data setting, or 2:1 splits for the 0.3% labelled-data setting (as in the 0.3% setting we have only 3 labels per class). Splits were chosen uniformly at random so that each class had an equal number of examples in the larger split, which was then used for training. Top-1 accuracy on the smaller split was used for hyperparameter tuning.

C.5. Dataset Transfer: Table 2

We found it important to recalculate the PMP rescaling matrix W_{PMP} on unlabeled data from the new dataset, and all hyperparameters were tuned on a 4:1 split of the training data. For Oxford Flowers, as there are only 1020 training images, which would result in a low-rank approximation to $W_{\text{PMP}} \in \mathbb{R}^{2048\times 2048}$, for each training image we generate 10 data augmented versions (using standard random crop and horizontal flips) to estimate the empirical covariance used in W_{PMP}. All datasets were obtained from the Torchvision PyTorch library (Paszke et al., 2019)

C.6. Different Architecture evaluation: Table 3

All experimental details follow those in Appendix C.4, and the pretrained ViT-B/16 checkpoint was again obtained from VISSL (Goyal et al., 2021).

D. Postman Pat

The Postman Pat abbreviation (which we further shorten to PMP) for our method, Post-hoc Manipulation of the Principal Axes & Trace, was inspired by the now retired British children's TV character: Postman Pat, pictured in Figure 11 with his cat Jess (though in the Danish version Jess is renamed to Emil).

Figure 11. Postman Pat and his cat Jess.