On the Stepwise Nature of Self-Supervised Learning

James B. Simon [1 2] **Maksis Knutins** [2] **Liu Ziyin** [3] **Daniel Geisz** [1] **Abraham J. Fetterman** [2] **Joshua Albrecht** [2]

Abstract

We present a simple picture of the training process of joint embedding self-supervised learning methods. We find that these methods learn their high-dimensional embeddings one dimension at a time in a sequence of discrete, well-separated steps. We arrive at this conclusion via the study of a linearized model of Barlow Twins applicable to the case in which the trained network is infinitely wide. We solve the training dynamics of this model from small initialization, finding that the model learns the top eigenmodes of a certain contrastive kernel in a stepwise fashion, and obtain a closed-form expression for the final learned representations. Remarkably, we then see the same stepwise learning phenomenon when training deep ResNets using the Barlow Twins, SimCLR, and VICReg losses. Our theory suggests that, just as kernel regression can be thought of as a model of supervised learning, *kernel PCA* may serve as a useful model of self-supervised learning.

1. Introduction

Self-supervised learning (SSL) has recently become a leading choice for representation learning using deep neural networks. Joint embedding methods, a prominent class of SSL methods, aim to ensure that any two "views" of the same input — for example, two random crops, or an image and its caption — are assigned similar representations. Such self-supervised approaches have yielded a bounty of recent empirical breakthroughs across domains (Hjelm et al., 2018; Wu et al., 2018; Bachman et al., 2019; He et al., 2020; Henaff, 2020; Chen & He, 2021; Radford et al., 2021; Caron et al., 2021; Assran et al., 2022b)

[1]UC Berkeley [2]Generally Intelligent [3]University of Tokyo. Correspondence to: James Simon <james.simon@berkeley.edu>. Code to reproduce results available at `https://gitlab.com/generally-intelligent/ssl_dynamics`.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the authors.

Despite SSL's simplicity, however, the field lacks a complete theoretical understanding of the paradigm. A promising place to start may be the fact that although there exist many different self supervised loss functions, these variants all achieve rather similar performance.[1] This similarity suggests that there may exist basic commonalities in their learning behavior which are as yet unidentified.

In this work, we propose a candidate for such a shared behavior. We present evidence that the high-dimensional embeddings of Barlow Twins (Zbontar et al., 2021), SimCLR (Chen et al., 2020), and VICReg (Bardes et al., 2021) are learned one dimension at a time in a series of discrete learning stages (see Figure 1). These embeddings are initially low-rank and increase in rank by one as each dimension is learned.

To reach this conclusion, we first study the Barlow Twins loss (Zbontar et al., 2021) applied to a linear model. The training of this model takes the form of a matrix factorization problem, and we present exact solutions to the training dynamics from small initialization. These solutions reveal that representation learning occurs in discrete, well-separated steps, each of which learns a top eigendirection of a certain contrastive kernel which characterizes the spectral bias of the model on the dataset. We also present a kernelization of our theory that allows its application to generic kernel machines, a class which importantly includes infinite-width neural networks.

Finally, we empirically examine the training of Barlow Twins, SimCLR, and VICReg using ResNets with various initializations and hyperparameters and in all cases clearly observe the stepwise behavior predicted by our analytical model. This behavior is most apparent upon small parameterwise initialization (Figures 1C,D and 2C-F) but persists even in realistic configurations (Figure 3). In light of this agreement, we suggest that SSL can be understood as a sequential process of learning orthogonal scalar functions in order of importance. This simple picture of SSL has the potential to both provide a useful lens for understanding the training behavior of self supervised networks and help guide

[1]For example, (Bardes et al., 2021) report that SimCLR, SwAV (Caron et al., 2020), Barlow Twins, VICReg, and BYOL (Grill et al., 2020) all score within a few percent of each other on ImageNet pretraining tasks.

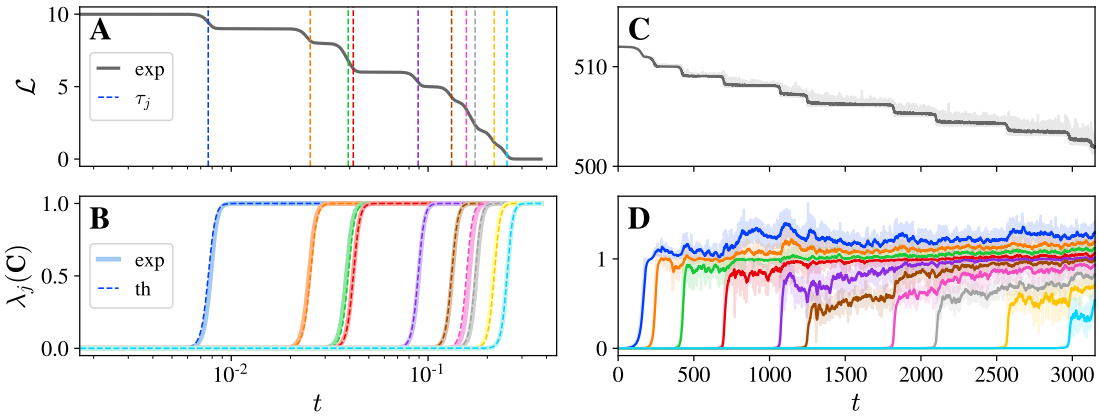

Figure 1: **SSL methods learn embeddings one dimension at a time in a series of discrete steps. (A,B)**: Loss and embedding eigenvalues for an analytical model of Barlow Twins using a linear model trained on $n = 500$ positive pairs from CIFAR-10. Solid curves show experimental trajectories, and dashed curves show predicted step timescales (A) and eigenvalues (B). Curves are plotted against effective time $t = [\text{lr}] \times [\text{step}]$. **(C,D)**: Loss and embedding eigenvalues for Barlow Twins using a deep ResNet on STL-10 with small initialization and $\lambda = 1$. Learning takes place in discrete, well-separated steps, each of which entails a drop in the loss and an increase in the dimensionality of the embeddings by one.

the design of new methods.

Concretely, our contributions are as follows:

- We propose a minimal model of Barlow Twins and solve its dynamics from near-zero initialization.
- We extend our model to arbitrary kernel machines including infinite-width neural networks and give closed-form equations for the final embeddings in terms of the training kernel.
- We validate our theory using Barlow Twins, VICReg and SimCLR with deep ResNets and find good qualitative agreement.

1.1. Motivation

Our theoretical approach is motivated by the infinite-width "neural tangent kernel" (NTK) limit of deep neural networks, in which the model's training dynamics become linear in its parameters (Jacot et al., 2018; Lee et al., 2019). This limit has been immensely fruitful for the study of ordinary supervised learning with deep neural networks, yielding a variety of useful insights and intuitions regarding training and generalization, many of which hold well even for the finite, highly nonlinear models used in practice.[2] Kernel regression with the NTK now serves as a useful simple model for supervised deep neural networks. Our aim here is to play the same trick with *self-supervised* learning, obtaining an analytical model that can be interrogated to answer questions about SSL training.

A further motivation is the fact that deep learning methods that learn flexible *representations* from unlabeled data, rather than a narrow target function from labeled data, are increasingly dominant in the modern deep learning ecosystem (Brown et al., 2020; Ramesh et al., 2022; Chowdhery et al., 2022). Despite this, many theoretical tools have been developed exclusively in the context of supervised learning. We aim to close this gap by porting theoretical tools from the study of supervised learning, such as kernel equivalences and spectral bias, over to the study of representation learning.

Similarly, perhaps the biggest open problem in the theory of supervised deep learning is understanding the process of *feature learning* at hidden layers. It is known that networks in the kernel limit do not adapt their hidden representations to data, so various works have explored avenues beyond the NTK, though none of these efforts have yet yielded simple closed-form equations for the features learned after training (Yang & Hu, 2021; Roberts et al., 2022; Bordelon & Pehlevan, 2022; Jacot et al., 2022). However, we find that networks in the kernel limit *do* learn nontrivial data embeddings when trained with SSL, and, what is more, we can obtain closed-form expressions for these final embeddings in terms of the NTK (Section 5). Our work presents an avenue by which feature and representation learning might be studied purely in terms of kernels.

[2] An incomplete list of examples includes Du et al. (2019); Cao et al. (2019); Yang (2019); Tancik et al. (2020); Fort et al. (2020); Yang & Hu (2021); Atanasov et al. (2021) for training and Mei & Montanari (2019); Allen-Zhu et al. (2019); Yang & Salman (2019); Arora et al. (2019b); Canatar et al. (2021); Simon et al. (2021); Mallinar et al. (2022); Loureiro et al. (2021); Wei et al. (2022) for generalization.

2. Related works

Theoretical study of SSL. Many recent works have sought to understand SSL through varied lenses including statistical learning theory (Arora et al., 2019c; Wei et al., 2020; Nozawa & Sato, 2021; Ash et al., 2021; HaoChen et al., 2021), information theory (Tsai et al., 2020; 2021; Tosh et al., 2021a;b), loss landscapes and training dynamics (Tian et al., 2020; Wang & Isola, 2020; Chen et al., 2021; Tian et al., 2021; Jing et al., 2021; Wen & Li, 2021; Pokle et al., 2022; Ziyin et al., 2022; Assran et al., 2022a), and kernel and spectral methods (Kiani et al., 2022; HaoChen et al., 2021; Johnson et al., 2022). Our work unifies dynamical and kernel perspectives of SSL pioneered by prior works as follows.

Ziyin et al. (2022) showed that most common SSL loss functions take the form $\mathrm{Tr}\left[\boldsymbol{W}\boldsymbol{A}\boldsymbol{W}^\top\right] + \mathrm{Tr}\left[(\boldsymbol{W}\boldsymbol{B}\boldsymbol{W}^\top)^2\right]$ for symmetric matrices A and B when expanded about the origin. This is the form of the loss function whose dynamics we solve to obtain exact optimization trajectories, and our exact solution can be adapted to other losses of the same form.

In influential work, HaoChen et al. (2021) showed that the optimal representation under a particular contrastive loss consists of the top Laplacian eigenmodes of a certain "augmentation graph" defined over the data space. However, as discussed by Saunshi et al. (2022), their approach is model-agnostic, and these optimal representations are usually highly degenerate in realistic cases.[3] This degeneracy must somehow be broken by the inductive bias of the model and optimization procedure. Our results complete this picture, characterizing this inductive bias for linearized SSL models as a bias towards small RKHS norm and thereby identifying which of the many zero-loss solutions is reached in practice.

In a similar spirit as our work, Balestriero & LeCun (2022), Kiani et al. (2022), and Cabannes et al. (2023) study SSL with linear models and kernel methods, deriving and comparing optimal representations for certain toy losses. They, too, find that kernelized SSL preferentially learns the top eigenmodes of certain operators, though their final representations differ from ours. One key difference is that these works jump to optima by solving for minimizers of the loss, whereas our focus is on the dynamics of training that lead to optimal representations. Since the loss often has many inequivalent global minimizers in practice (as is indeed the case for Barlow Twins), understanding these dynamics is necessary for determining *which* global minimum will actually be found by gradient descent.

Matrix factorization and deep linear networks. Our minimal model of Barlow Twins turns out to be closely related to classic matrix factorization problems (Gunasekar et al., 2017; Li et al., 2018; Arora et al., 2019a; Chi et al., 2019). For example, our Equation 5 resembles Equation 3.1 of Li et al. (2018). Matrix factorization problems are often studied as paradigmatic examples of tasks with many inequivalent zero-loss solutions, with the challenge then to characterize the model's inductive bias and understand which of these solutions is actually reached by gradient descent. This is often doable under an assumption of small initialization, from which gradient descent finds the solution which minimizes a certain matrix norm (Gunasekar et al., 2017; Li et al., 2018), and this is indeed the case in our model. Our work draws a new link between SSL and matrix factorization, and we propose that degenerate matrix factorization problems can serve as illuminating simplified models for representation learning tasks more broadly.

The dynamics of our analytical model (Proposition 4.1) bear a striking resemblance to the dynamics of deep linear networks.[4] (Fukumizu, 1998; Saxe et al., 2013; Du et al., 2018; Arora et al., 2018; Jacot et al., 2021) Deep linear networks initialized near zero also exhibit learning in discrete, well-separated stages, each of which entails the learning of one singular direction in the model function[5]Furthermore, in both cases, the loss plateaus between consecutive learning stages occur when the model passes near saddle points, with the index of the saddle point decreasing by one each stage (Jacot et al., 2021). However, the problems differ in key ways: in our setting, the stagewise learning behavior is a result of the self-supervised loss function, not the depth of the network, and our problem has many inequivalent global minima, unlike the typical deep linear setting.

Physics and symmetry breaking. Interestingly, Landau (1944) encountered the same differential equation we find for embedding eigencoefficients in the study of the onset of turbuluent fluid flow. This is a consequence of the fact that both representation learning and the onset of turbulence are processes of spontaneous symmetry breaking. We make this connection explicitly in Appendix F.

3. Preliminaries

We will study an ordinary linear model trained in a contrastive fashion. Suppose we have a dataset of n "positive pairs" $\boldsymbol{x}_i, \boldsymbol{x}'_i \in \mathbb{R}^m$ for $i \in 1...n$.[6] Our model will consist

[3]Specifically, in order to be nonvacuous, their approach requires that it is common that a particular image will appear as an augmentation of *multiple different base images* so that the augmentation graph is largely connected. This will virtually never happen with realistic dataset sizes and step counts.

[4]Note that the training of a deep linear network is itself an asymmetric matrix factorization problem.

[5].

[6]We use a finite dataset of n positive pairs for simplicity, but our theory works equally well when optimizing on the population

of a linear transformation to a d-dimensional embedding parameterized as $\boldsymbol{f}(\boldsymbol{x}) \equiv \boldsymbol{W}\boldsymbol{X}$ with $\boldsymbol{W} \in \mathbb{R}^{d\times m}$. We would like to learn a transformation $\boldsymbol{W}$ such that positive pairs have similar embeddings but the full set of embeddings maintains high diversity.

To encourage such representations, Barlow Twins prescribes a loss function which pushes the cross-correlation matrix between $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{x}')$ towards the identity matrix. We will use a slightly simplified variant of the Barlow Twins loss given by

$$\mathcal{L} = \left\|\boldsymbol{C} - \boldsymbol{I}_d\right\|_F^2, \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm and $\boldsymbol{C} \in \mathbb{R}^{d\times d}$ is the cross-correlation matrix given by

$$\boldsymbol{C} \equiv \frac{1}{2n}\sum_i \left(\boldsymbol{f}(\boldsymbol{x}_i)\boldsymbol{f}(\boldsymbol{x}_i')^\top + \boldsymbol{f}(\boldsymbol{x}_i')\boldsymbol{f}(\boldsymbol{x}_i)^\top\right). \tag{2}$$

Compared to the original loss of (Zbontar et al., 2021), we have set the hyperparameter λ to 1, removed batchnorm in the definition of $\boldsymbol{C}$, and symmetrized $\boldsymbol{C}$.

We will initialize $\boldsymbol{W} = \boldsymbol{W}_0$ and train with full-batch gradient flow as

$$\frac{d\boldsymbol{W}}{dt} = -\nabla_{\boldsymbol{W}}\mathcal{L}. \tag{3}$$

We wish to understand both the dynamics of this training trajectory and the final weights $\boldsymbol{W}_\infty \equiv \lim_{t\to\infty}\boldsymbol{W}$.

4. Solving the dynamics of the linear model

4.1. The feature cross-correlation matrix $\boldsymbol{\Gamma}$

The task we have set up is a matrix optimization problem. To elucidate the structure of this problem, we can simplify $\boldsymbol{C}$ as

$$\boldsymbol{C} = \frac{1}{2n}\sum_i \boldsymbol{W}\left(\boldsymbol{x}_i\boldsymbol{x}_i'^\top + \boldsymbol{x}_i'\boldsymbol{x}_i^\top\right)\boldsymbol{W}^\top = \boldsymbol{W}\boldsymbol{\Gamma}\boldsymbol{W}^\top \tag{4}$$

where we have defined the feature cross-correlation matrix $\boldsymbol{\Gamma} \equiv \frac{1}{2n}\sum_i(\boldsymbol{x}_i\boldsymbol{x}_i'^\top + \boldsymbol{x}_i'\boldsymbol{x}_i^\top)$. Equation 1 then becomes

$$\mathcal{L} = \left\|\boldsymbol{W}\boldsymbol{\Gamma}\boldsymbol{W}^\top - \boldsymbol{I}_d\right\|_F^2, \tag{5}$$

a form reminiscent of matrix factorization problems, and Equation 3 is

$$\frac{d\boldsymbol{W}}{dt} = -4\left(\boldsymbol{W}\boldsymbol{\Gamma}\boldsymbol{W}^\top - \boldsymbol{I}_d\right)\boldsymbol{W}\boldsymbol{\Gamma}. \tag{6}$$

We will denote by $\gamma_1 \geq \ldots \geq \gamma_m$ the eigenvalues of $\boldsymbol{\Gamma}$ and, for any $k \in 1\ldots m$, let $\boldsymbol{\Gamma}^{(\leq k)} \in \mathbb{R}^{k\times m}$ be the matrix containing the top k eigenvectors of $\boldsymbol{\Gamma}$ as rows.

loss of a data distribution, which simply corresponds to the case $n \to \infty$.

4.2. Exact solutions for aligned initialization

It is nontrivial to solve Equation 6 from arbitrary initialization. However, as is common in matrix factorization problems, we can obtain exact trajectories starting from special initializations, and these special solutions will shed light on the general dynamics. We first consider an "aligned initialization" in which the right singular vectors of $\boldsymbol{W}_0$ are the top eigenvectors of $\boldsymbol{\Gamma}$. Concretely, let

$$\boldsymbol{W}_0 = \boldsymbol{U}\boldsymbol{S}_0\boldsymbol{\Gamma}^{(\leq d)} \tag{7}$$

be the singular value decomposition of $\boldsymbol{W}_0$ with $\boldsymbol{U} \in \mathbb{R}^{d\times d}$ an arbitrary orthonormal matrix, and $\boldsymbol{S}_0 \in \mathbb{R}^{d\times d}$ is a matrix of singular values given by

$$\boldsymbol{S}_0 = \text{diag}(s_1(0), ..., s_d(0)) \tag{8}$$

with $s_j(0) > 0$.[7] The dynamics of $\boldsymbol{W}(t)$ under Equation 6 are then given by the following Proposition:

Proposition 4.1 (Trajectory of $\boldsymbol{W}(t)$ from aligned initialization)**.** *If $\boldsymbol{W}(0) = \boldsymbol{W}_0$ as given by Equations 7 and 8, then*

$$\boldsymbol{W}(t) = \boldsymbol{U}\boldsymbol{S}(t)\boldsymbol{\Gamma}^{(\leq d)} \tag{9}$$

with $\boldsymbol{S}(t) = diag(s_1(t), ..., s_d(t))$ and

$$s_j(t) = \frac{e^{4\gamma_j t}}{\sqrt{s_j^{-2}(0) + (e^{8\gamma_j t} - 1)\gamma_j}}. \tag{10}$$

Proof of Proposition 4.1. Treating Equation 9 as an ansatz and inserting it into Equation 6, we find that

$$\frac{d\boldsymbol{W}}{dt} = 4\boldsymbol{U}(1 - \boldsymbol{D}\boldsymbol{S}(t)^2)\boldsymbol{D}\boldsymbol{S}(t)\boldsymbol{\Gamma}^{(\leq d)}, \tag{11}$$

with $\boldsymbol{D} = \text{diag}(\gamma_1, \ldots, \gamma_d)$. It follows that the singular vectors of $\boldsymbol{W}(t)$ remain fixed, and the singular values evolve according to

$$s_j'(t) = 4\left(1 - \gamma_j s_j^2(t)\right)\gamma_j s(t). \tag{12}$$

This ODE can be solved to yield Equation 10. □

When $t \to \infty$, Proposition 4.1 prescribes singular values equal to

$$s_j(\infty) = \begin{cases} \gamma_j^{-1/2} & \text{for } \gamma_j > 0, \\ s_j(0) & \text{for } \gamma_j = 0, \\ 0 & \text{for } \gamma_j < 0. \end{cases} \tag{13}$$

Each singular value thus flows monotonically towards either $\gamma_j^{-1/2}$ or zero depending on whether the corresponding

[7] We assume alignment with the top d eigenvectors both for notational simplicity and because this is the solution we will ultimately care about, but our exact solution will hold for any set of d eigenvectors of $\boldsymbol{\Gamma}$.

eigenvalue is positive or negative. This can be understood by noting that the loss (Equation 5) can be rewritten as

$$\mathcal{L} = \sum_j (1 - \gamma_j s_j^2)^2, \tag{14}$$

which makes clear that if $\gamma_j > 0$, then $s_j = \gamma_j^{-1/2}$ is optimal (and achieves zero loss on the j-th term of the sum), but if $\gamma_j < 0$, then the model can do no better than $s_j = 0$

It is worth noting that $\lambda_j = \gamma_j s_j^2$ is the corresponding eigenvalue of $\boldsymbol{C}$. With this change of coordinates, the trajectories of Proposition 4.1 become nearly sigmoidal, with $\lambda_j \approx (1 + \lambda_j^{-1}(0) e^{-8\gamma_j t})^{-1}$ when $|\lambda_j(0)| \ll 1$.

We will be particularly interested in the set of terminal solutions. Accordingly, let us define the set of *top spectral* $\boldsymbol{W}$ as follows:

Definition 4.2. (Top spectral $\boldsymbol{W}$) A top spectral $\boldsymbol{W}$ is one for which $\boldsymbol{W} = \boldsymbol{U}\tilde{\boldsymbol{S}}\boldsymbol{\Gamma}^{(\leq d)}$, with $\boldsymbol{U}$ an orthogonal matrix and $\tilde{\boldsymbol{S}} = \text{diag}(\gamma_1^{-1/2}\mathbb{1}_{\gamma_1>0}, ..., \gamma_d^{-1/2}\mathbb{1}_{\gamma_d>0})$.

(Note that these are precisely the set of $\boldsymbol{W}(\infty)$ found by Proposition 4.1 save for the edge case $\gamma_j = 0$, in which case we set $s_j = 0$.) These solutions form an equivalence class parameterized by the rotation matrix $\boldsymbol{U}$. As observed by HaoChen et al. (2021), such a rotation makes no difference for the downstream generalization of a linear probe, so we may indeed view all top spectral $\boldsymbol{W}$ as equivalent.

Let us assume henceforth that $\gamma_1, ..., \gamma_d > 0$ and $\gamma_d > \gamma_{d+1}$. The top spectral $\boldsymbol{W}$ achieve $\mathcal{L}(\boldsymbol{W}) = 0$, but note that there generally exist other optima, such as those aligned with a different set of positive eigenvectors. However, the top spectral $\boldsymbol{W}$ are optimal in the following sense:

Proposition 4.3. *The top spectral $\boldsymbol{W}$ are precisely the solutions to*

$$\underset{\boldsymbol{W}}{\text{argmin}} \|\boldsymbol{W}\|_F \quad \text{s.t.} \quad \mathcal{L}(\boldsymbol{W}) = 0. \tag{15}$$

We relegate the proof to Appendix C. Proposition 4.3 implies that, of all solutions achieving $\mathcal{L} = 0$, the top spectral solutions have minimal $\|\boldsymbol{W}\|_F$. Noting that gradient descent often has an implicit bias towards low-norm solutions (Gunasekar et al., 2017; Soudry et al., 2018), we might expect to reach this set of optima from some more general initial conditions.

4.3. The case of small initialization

Returning to Proposition 4.1, an informative special case of our exact dynamical solution is that in which the initial singular values are small relative to their final values ($s_j(0) \ll \gamma_j^{-1/2}$). In this case, Equation 10 states that $s_j(t)$ will remain small up to a critical time

$$\tau_j = \frac{-\log(s_j^2(0)\gamma_j)}{8\gamma_j} \tag{16}$$

after which it will rapidly grow to its final value. Note that τ_j is principally set by γ_j , with only a weak logarithmic dependence on initialization $s_j(0)$. The learning dynamics can thus be understood as a *stepwise process* in which d orthogonal directions are each rapidly learned at their respective timescales τ_j, with plateaus in between adjacent learning phases.

Proposition 4.1 assumed a special aligned initialization for $\boldsymbol{W}$. We will now give a result which generalizes this significantly, showing that the trajectory from any *small* initialization closely follows that from a particular aligned initialization.

In order to state our result, we will first define the QR factorization and an "alignment transformation."

Definition 4.4 (QR factorization.)**.** The *QR factorization* of a matrix $\boldsymbol{M} \in \mathbb{R}^{a\times b}$ returns a decomposition $\boldsymbol{QR} = \boldsymbol{M}$ such that $\boldsymbol{Q} \in \mathbb{R}^{a\times a}$ is orthogonal and $\boldsymbol{R} \in \mathbb{R}^{a\times b}$ is upper-triangular with nonnegative diagonal. If $a \leq b$ and $\boldsymbol{M}$ is full rank, then the QR factorization is unique.

Definition 4.5 (Alignment transformation.)**.** The *alignment transformation* $\mathcal{A}$ of a matrix $\boldsymbol{M}$ returns a matrix $\mathcal{A}(\boldsymbol{M}) = \boldsymbol{Q}\tilde{\boldsymbol{R}}$, where $\boldsymbol{QR} = \boldsymbol{M}$ is a QR factorization and $\tilde{\boldsymbol{R}}$ is $\boldsymbol{R}$ with all off-diagonal elements set to zero.

We can now state the main result of this section.

Result 4.6 (Trajectory from generic small initialization)**.**

- *Let $\gamma_1, ..., \gamma_d$ be unique.*
- *Let $\tilde{\boldsymbol{W}}_0 \in \mathbb{R}^{d\times m}$ with $\tilde{\boldsymbol{W}}_0\boldsymbol{\Gamma}^{(\leq d)}$ full rank.*
- *Let $\boldsymbol{W}(t)$ be the solution to Equation 6 with initial condition $\boldsymbol{W}(0) = \alpha\tilde{\boldsymbol{W}}_0$.*
- *Let $\boldsymbol{W}^*(t)$ be the aligned solution with initial condition $\boldsymbol{W}^*(0) = \mathcal{A}(\boldsymbol{W}(0)\boldsymbol{\Gamma}^{(\leq m)\top})\boldsymbol{\Gamma}^{(\leq m)}$.*

Then as $\alpha \to 0$, $\|\boldsymbol{W}(t) - \boldsymbol{W}^*(t)\|_F \to 0$ for all t.

We give a derivation of this result in Appendix C.[8]

Result 4.6 states that the trajectory from generic small initialization closely follows a particular aligned trajectory. This aligned trajectory is given in closed form by Proposition 4.1, and so this result gives us equations for the dynamics from arbitrary initialization.

[8]We style this conclusion as a Result rather than a Theorem because we give an informal derivation rather than a formal proof. We conjecture that this result can indeed be made formal.

Some intuition for this result can be gained by examining the construction of $\boldsymbol{W}^*(0)$. The aligned solution $\boldsymbol{W}^*(t) = \boldsymbol{U}\boldsymbol{S}^*(t)\boldsymbol{\Gamma}^{(\leq d)}$ is composed solely of the top d eigendirections of $\boldsymbol{\Gamma}$, but an arbitrary initialization will have no such preference. How does this alignment occur? Note that, at early times when $\boldsymbol{W}$ is small, the quadratic term of the loss will dominate, and Equation 6 reduces to

$$\frac{d\boldsymbol{W}}{dt} \approx 4\boldsymbol{W}\boldsymbol{\Gamma} \quad \Rightarrow \quad \boldsymbol{W} \approx \boldsymbol{W}(0)\, e^{4\boldsymbol{\Gamma}t}. \tag{17}$$

The top eigendirections of $\boldsymbol{\Gamma}$ grow faster and will quickly dominate, and after a time $\tilde{\tau} \gg (\gamma_d - \gamma_{d+1})^{-1}$, we will have

$$\boldsymbol{W} \approx \boldsymbol{W}(0)\boldsymbol{\Pi}^{(\leq d)}\, e^{4\boldsymbol{\Gamma}t} \tag{18}$$

where $\boldsymbol{\Pi}^{(\leq d)} \equiv \boldsymbol{\Gamma}^{(\leq d)\top}\boldsymbol{\Gamma}^{(\leq d)}$ is the projector onto the top-d eigenspace of $\boldsymbol{\Gamma}$. Components aligned with eigenmodes of index $j > d$ are thus negligible compared to those of index $\leq d$ and do not interfere in the learning dynamics, which converge before such later eigenmodes can grow to order one.

Having identified the relevant eigendirections, we must now determine their effective initial singular values $s_j^*(0)$. Let $\boldsymbol{v}_j$ be the j-th eigenvector of $\boldsymbol{\Gamma}$ with $\|\boldsymbol{v}_j\| = 1$ and define $\boldsymbol{u}_j = \boldsymbol{W}(0)\boldsymbol{v}_j$. If $\boldsymbol{v}_j$ were a right singular vector of $\boldsymbol{W}(0)$, we would have $s_j^*(0) = \|\boldsymbol{u}_j\|$. We will not be so fortunate in general, however. Examining Equation 6, we should expect each eigenmode to only be able to grow in the subspace of $\mathbb{R}^d$ which has not already been filled by earlier eigenmodes, which suggests that we take

$$s_j^*(0) = \left\| \left(1 - \sum_{k<j} \frac{\boldsymbol{u}_k^\top \boldsymbol{u}_k}{\|\boldsymbol{u}_k\|^2} \right) \boldsymbol{u}_j \right\|. \tag{19}$$

These are precisely the singular values of $\boldsymbol{W}^*(0)$.[9]

4.4. Numerical simulation

We perform basic numerical simulations of our linear problem which verify Proposition 4.1 and Result 4.6. We sample $n = 500$ random images from CIFAR-10 (Krizhevsky, 2009) and, for each, take two random crops to size $20 \times 20 \times 3$ to obtain n positive pairs (which thus have feature dimension $m = 1200$). We then randomly initialize a linear model with output dimension $d = 10$ and weights drawn i.i.d. from $\mathcal{N}(0, \alpha^2)$ with $\alpha = 10^{-7}$ and train with the loss of Equation 1 with learning rate $\eta = 5 \times 10^{-5}$.

During training, we track both the loss and the eigenvalues of the embedding cross-correlation matrix $\boldsymbol{C}$. Our stepwise dynamics predict the loss will start at $\mathcal{L}(0) \approx d$ and decrease by one as each mode is learned, giving

$$\mathcal{L}(t) \approx \sum_{j|\tau_j > t} 1. \tag{20}$$

The eigenvalues of $\boldsymbol{C}$ will be $(\gamma_1 s_1^2(t), ..., \gamma_d s_d^2(t))$, with $s_j(t)$ given by Proposition 4.1. The use of Proposition 4.1 requires values for $s_1(0), ..., s_d(0)$, and these can be found from Equation 19 and the statistics of the random initialization to be roughly

$$s_j(0) \approx \sigma\sqrt{d - j + 1}. \tag{21}$$

The results are plotted in Figure 1(A,B). We find excellent agreement with our theory.

5. Kernelizing the linear model

Our discussion has so far dealt with an *explicit* linear regression problem in which we have direct access to the data features $\boldsymbol{x}_i$. We used this explicit representation to construct $\boldsymbol{\Gamma} \in \mathbb{R}^{m\times m}$, which lives in feature space. However, many models of interest are linear with an *implicit* feature space, with the output a linear function not of the input x but rather of a fixed transformation of the input $\boldsymbol{\phi}(x)$. Models of this class include kernel machines (Shawe-Taylor et al., 2004), random feature models and deep neural networks in which only the last layer is trained (Rahimi et al., 2007; Lee et al., 2018), and infinite-width neural networks, the latter of which evolve under the NTK (Jacot et al., 2018). While these models are equivalent to linear models, we do not have an explicit representation of the features $\boldsymbol{\phi}(x)$ (which may be large or even infinite-dimensional). What we *do* have access to is the model's *kernel function* $\mathcal{K}(x, x') = \boldsymbol{\phi}(x)^\top \boldsymbol{\phi}(x')$.

In our original linear model, the kernel is simply the inner product $\boldsymbol{x}^\top \boldsymbol{x}'$. Reinterpreting $\boldsymbol{x}$ as $\boldsymbol{\phi}(x)$, any quantity expressed solely in terms of such inner products can still be computed after kernelization. Our challenge, then, is to rewrite our theory of dynamics entirely in terms of such inner products so we may apply it to general kernel machines.

5.1. Kernelized solution

We will manage to do so. Let our datasets be $\mathcal{X} \equiv \{x_1, ..., x_n\}$ and $\mathcal{X}' \equiv \{x'_1, ..., x'_n\}$. Let us define the kernel matrix $\boldsymbol{K}_{\mathcal{X}\mathcal{X}} \in \mathbb{R}^{n\times n}$ such that $[\boldsymbol{K}_{\mathcal{X}\mathcal{X}}]_{ij} = \mathcal{K}(x_i, x_j)$ and define $\boldsymbol{K}_{\mathcal{X}\mathcal{X}'}$, $\boldsymbol{K}_{\mathcal{X}'\mathcal{X}}$, and $\boldsymbol{K}_{\mathcal{X}'\mathcal{X}'}$ analogously. Let us also define

$$\tilde{\boldsymbol{K}} \equiv \begin{bmatrix} \boldsymbol{K}_{\mathcal{X}\mathcal{X}} & \boldsymbol{K}_{\mathcal{X}\mathcal{X}'} \\ \boldsymbol{K}_{\mathcal{X}'\mathcal{X}} & \boldsymbol{K}_{\mathcal{X}'\mathcal{X}'} \end{bmatrix}, \tag{22}$$

the kernel over the combined dataset, as well as

$$\boldsymbol{Z} \equiv \frac{1}{2n} \begin{bmatrix} \boldsymbol{K}_{\mathcal{X}\mathcal{X}'}\boldsymbol{K}_{\mathcal{X}\mathcal{X}} & \boldsymbol{K}_{\mathcal{X}\mathcal{X}'}^2 \\ \boldsymbol{K}_{\mathcal{X}'\mathcal{X}'}\boldsymbol{K}_{\mathcal{X}\mathcal{X}} & \boldsymbol{K}_{\mathcal{X}'\mathcal{X}'}\boldsymbol{K}_{\mathcal{X}\mathcal{X}'} \end{bmatrix} + [\text{transpose}]. \tag{23}$$

[9] This can be seen by noting that the QR factorization involves the Gram-Schmidt-like process of Equation 19.

Finally, let us define $\boldsymbol{K}_{\boldsymbol{\Gamma}} \equiv \tilde{\boldsymbol{K}}^{-1/2}\boldsymbol{Z}\tilde{\boldsymbol{K}}^{-1/2} \in \mathbb{R}^{2n\times 2n}$, where $\tilde{\boldsymbol{K}}^{-1/2}$ is interpreted as $(\tilde{\boldsymbol{K}}^{+})^{-1/2}$ if $\tilde{\boldsymbol{K}}$ is degenerate. The matrix $\boldsymbol{K}_{\boldsymbol{\Gamma}}$ is symmetric and akin to a kernelized version of $\boldsymbol{\Gamma}$.[10][11]

Let $(\gamma_j, \boldsymbol{b}_j)$ be the eigenvalues and normalized eigenvectors of $\boldsymbol{K}_{\boldsymbol{\Gamma}}$ indexed in descending eigenvalue order. The kernelization of our theory is given by the following proposition.

Proposition 5.1 (Kernelized solution)**.**

(a) All nonzero eigenvalues of $\boldsymbol{K}_{\boldsymbol{\Gamma}}$ are eigenvalues of $\boldsymbol{\Gamma}$ and vice versa.

(b) The top spectral solution gives the embeddings

$$\boldsymbol{f}(x) = \boldsymbol{U}\tilde{\boldsymbol{S}}[\boldsymbol{b}_1 \dots \boldsymbol{b}_d]^\top \tilde{\boldsymbol{K}}^{-1/2}[\boldsymbol{K}_{x\mathcal{X}}\ \boldsymbol{K}_{x\mathcal{X}'}]^\top \quad (24)$$

with $\boldsymbol{U}$ an orthogonal matrix, $\tilde{\boldsymbol{S}} = diag(\gamma_1^{-1/2}, ..., \gamma_d^{-1/2})$, and $\boldsymbol{K}_{x\mathcal{X}}, \boldsymbol{K}_{x\mathcal{X}'} \in \mathbb{R}^{1\times n}$ such that $[\boldsymbol{K}_{\mathcal{X}x}]_i = \mathcal{K}(x_i, x)$ and $[\boldsymbol{K}_{\mathcal{X}'x}]_i = \mathcal{K}(x'_i, x)$.

(c) The top spectral solutions correspond to the embeddings $\boldsymbol{f}(x)$ given by

$$\underset{\boldsymbol{f}}{argmin}\, \|\boldsymbol{f}\|_{\mathcal{K}} \quad s.t. \quad \mathcal{L}(\boldsymbol{f}) = 0. \quad (25)$$

We give the proof and an expression for $\boldsymbol{f}(x,t)$, the embeddings over time from small (aligned) initialization, in Appendix B. These results allow us to predict both the training dynamics and final embeddings of our contrastive learning problem with a black-box kernel method.

5.2. Implications of kernelized solution

This kernelized solution has several interesting properties that we will briefly discuss here.

Special case: $\mathcal{X} = \mathcal{X}'$. In the case in which the two views of the data are identical, one finds that $\boldsymbol{K}_{\boldsymbol{\Gamma}} = \tilde{\boldsymbol{K}}$ and the model simply learns the top eigenmodes of its base (neural tangent) kernel.

SSL as kernel PCA. Proposition 5.1(b) states that the final embeddings are governed by the top d eigenvectors of the kernel-like matrix $\boldsymbol{K}_{\Gamma}$. With our setup, then, **SSL with neural networks amounts to *kernel PCA* in the infinite-width limit.** This is analogous to the fact that standard supervised learning approaches kernel *regression* in the infinite-width limit.[12] This is rather satisfying in light of the fact that kernel regression and kernel PCA are the simplest supervised and unsupervised kernel methods, respectively.

The same theory works for multimodal SSL. The above solution holds for the infinite-width NTK limit of any neural network architecture. It is worth noting that this includes multimodal setups such as CLIP (Radford et al., 2021) in which representations for the two datasets $\mathcal{X}$ and $\mathcal{X}'$ are generated by *different* models, and the two datasets may be of different types. We can view the two models as one combined model with two pathways, and since these pathways share no parameters, we have $\boldsymbol{K}_{\mathcal{X}\mathcal{X}'} = \boldsymbol{K}_{\mathcal{X}'\mathcal{X}} = \boldsymbol{0}$.[13] Both our linear and kernelized solutions remain valid and nonvacuous in this setting.

Generalization on downstream tasks. The quality of a learned representation is often assessed by fitting a downstream function g^* (such as an image classification) with a linear function of the representation as $\hat{g}(x) = \boldsymbol{\beta}^\top \boldsymbol{f}(x)$. Downstream task performance will be good if g^* lies largely in the linear span of the components of $\boldsymbol{f}(x)$. Since Proposition 5.1(b) yields $\boldsymbol{f}(x)$ in closed form, generalization can thus be assessed in our setting. We leave this direction for future work.

Mapping from initial to final kernel. Downstream task performance will be determined by the learned kernel $\mathcal{K}_{\text{emb}}(x, x') \equiv \boldsymbol{f}(x)^\top \boldsymbol{f}(x')$, which contains all the information of $\boldsymbol{f}$ save for the arbitrary rotation $\boldsymbol{U}$. We can thus think of SSL as a process which maps an initial, naive kernel $\mathcal{K}$ to a final kernel $\mathcal{K}_{\text{emb}}$ which has learned the structure of the data. Many other poorly-understood processes in deep learning — most notably that of feature learning — also have the type signature `(initial kernel, data)` $\rightarrow$ `(final kernel)`, but there exist few closed-form algorithms with this structure. While representation learning and supervised feature learning are different processes, it seems likely that they will prove to be related,[14] and thus our closed-form solution for the final kernel may be useful for the study of feature learning.

6. Experiments

Since our theory was derived in the case of linear models, a natural question is whether it is useful and informative even for the study of practical deep neural networks. **Here we present evidence that the stepwise learning phenomenon**

[10] Here $\boldsymbol{\Gamma} = \frac{1}{2n}\sum_i \left(\boldsymbol{\phi}(x_i)\boldsymbol{\phi}(x'_i)^\top + \boldsymbol{\phi}(x'_i)\boldsymbol{\phi}(x_i)^\top\right)$ since we have reinterpreted $\boldsymbol{x}$ as $\boldsymbol{\phi}(x)$. This $\boldsymbol{\Gamma}$ would be sufficient to use our theory of Section 4 were it not constructed of *outer* products of $\boldsymbol{\phi}(x_i)$'s, which are inaccessible after kernelization.

[11] While $\boldsymbol{K}_{\boldsymbol{\Gamma}}$ is not technically a kernel matrix as it can have negative eigenvalues, we may heuristically think of it as one since only the (top) positive eigenvalues typically matter.

[12] This analogy can be furthered by noting that these equivalences both occur as $t \rightarrow \infty$ and both require small or zero initialization.

[13] In the parlance of the original linear model, the feature vectors of $\mathcal{X}$ and $\mathcal{X}'$ lie in orthogonal subspaces, and the model is tasked with discovering correlations across these subspaces.

[14] See Geirhos et al. (2020) and Grigg et al. (2021) for evidence in this direction.

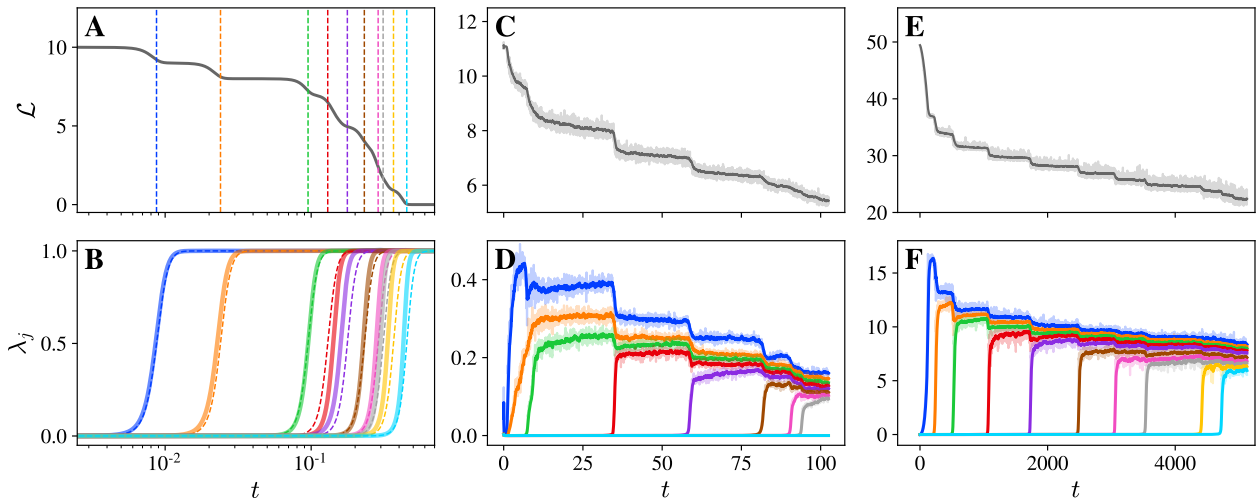

Figure 2: **Stepwise learning in SSL with nonlinear neural networks trained from small initialization.** Losses and embedding eigenvalues as a function of time $t = [\text{lr}] \times [\text{step}]$ for **(A, B)** a single-hidden-layer MLP trained with our simplified Barlow Twins loss, **(C, D)** a ResNet trained with SimCLR loss, and **(E, F)** a ResNet trained with VICReg loss, all trained on STL-10.

we identified occurs even in realistic SSL setups with ResNet-50 encoders, and even with losses besides Barlow Twins. We sketch experiments here and provide more details in Appendix D. In all figures, reported eigenvalues are those of the cross-correlation matrix $\boldsymbol{C}$ when the loss is Barlow Twins, and are those of the ordinary covariance matrix of the embeddings when the loss is SimCLR or VICReg unless otherwise stated, though in practice these give quite similar results.

We perform a series of experiments with increasingly realistic models. As a first nonlinear experiment, we revisit our previous numerical simulation of our linear model (Section 4.4) and simply replace the model with a width-2048 ReLU MLP with a single hidden layer using the standard PyTorch parameterization. This model is fairly wide, and so we expect its NTK to remain roughly fixed. Results are shown in Figure 2 (A,B).

We next train practical deep SSL methods with some hyperparameters modified slightly so as to better align with our theory. We train VICReg, SimCLR, and Barlow Twins using a ResNet-50 encoder and an MLP projection head on the full STL-10 dataset (Coates et al., 2011). These runs use small parameterwise initialization and small learning rate, but important hyperparameters such as momentum and weight decay are otherwise kept largely as in the original publications[15]. The results, plotted in Figure 1(C,D) (Barlow Twins) and Figure 2(C-F) (SimCLR and VICReg), clearly show the same stepwise learning behavior seen in simpler settings[16]. Agreement with predictions using the initial NTK is poor as expected from work on ResNet NTKs (Fort et al., 2020; Vyas et al., 2022) and we omit theory curves.

Finally, we repeat these experiments in fully realistic settings, with standard initialization and learning rates. Even though eigenvalues are no longer uniformly small at initialization, we still see stepwise learning in their growth, as shown in Figure 3. Specifically, we see for each loss that eigenvalues separate clearly into a band of eigenvalues that have not yet grown and a band of those that have fully grown with a sparse region in between, and eigenvalues move from the lower band to the upper band as training proceeds. This interpretation is affirmed by eigenvalue histograms throughout training, which reveal bimodal distributions as shown in Figure 4. Our view is that stepwise learning — i.e., the sequential growth of the rank of the embeddings, corresponding to the learning of orthogonal functions — is generic, and is simply made cleaner upon taking small initialization[17].

Stepwise behavior in hidden representations. All theory and experiments thus far have studied the final embeddings of the network. However, in practice, one typically uses the hidden representations of the network taken several layers

[15]We also changed batch size for practical reasons related to calculating the kernel and set $\lambda = 1$ for Barlow Twins. See Appendix D for full details.

[16]Unlike with the three reported methods, we did not immediately see stepwise learning with BYOL.

[17]Figure 3 suggests that, at least for Barlow Twins and VICReg, embedding directions may be learned "a few at a time" rather than one at a time with realistic hyperparameters, but when the embedding dimension is also realistically large, this difference is minor.

before the output for downstream tasks. In light of this, we repeat our experiments with small initialization but measure eigenvalues computed from the hidden representations. Results are reported in Figure 5. Remarkably, despite the fact that our theory says nothing about hidden representations, *we still clearly see learning steps coinciding with the learning steps at the embeddings.* Understanding the couplings in these dynamics is an interesting direction for future work.

Theoretical predictions of embeddings with the empirical NTK. Our theory predicts not only the occurrence of learning steps in the model embeddings but also the precise embedding function learned (up to an orthogonal transformation) in terms of the model's NTK. As a final experiment, we compare these predictions, evaluated using the empirical NTK after training, with the true embeddings learned in ResNet-50 experiments from small initialization with $d = 50$. The subspaces spanned by the empirical and predicted embeddings have alignment significantly above chance, with the theoretical predictions capturing a bit over 50% of the span of the true embeddings in a precise sense we define.

Intriguingly, we also find that the embeddings have comparable agreement *between* SSL methods. This suggests that these methods are in fact learning similar things, which is encouraging for the prospects for developing unifying theory for SSL. We give details of these experiments in Appendix E.

7. Conclusions

We have presented and solved a linearized minimal model of Barlow Twins (Sections 4 and 5). Our solution reveals that, as training proceeds, the model sequentially learns the top d eigenmodes of a certain kernel in a discrete fashion, stopping when the embeddings are full rank. Turning to bona fide ResNets in near-realistic training settings, we see precisely this learning behavior for Barlow Twins, SimCLR, and VICReg in both their embeddings and hidden representations. This paints a new, useful picture of the training dynamics of SSL: rather than a black-box optimization process that magically converges on final representations, we can perhaps think of self-supervised training as an iterative process of selecting desirable rank-one functions and tacking them onto a growing representation.

Our theory has several clear limitations. First, practical deep neural networks are *not* kernel methods: the NTK is known to change over time (Yang & Hu, 2021; Vyas et al., 2022). This suggests that our theory's predicted representations will not closely match those predicted by the initial NTK in practical cases, though it is plausible that they *will* match those predicted by the empirical NTK *after* training in light of similar results for supervised learning (Long, 2021; Atanasov et al., 2021). Second, in practice, downstream tasks in self-supervised pipelines are usually trained not on the final embeddings of the SSL model but rather on the hidden representation some layers prior, while our theory (like virtually all current SSL theory) only describes the embeddings. We partially surmount this limitation by empirically observing stepwise behavior in representations, though developing theory for this observation appears a worthwhile pursuit. We note that it is not currently known why the use of hidden representations is preferable, but having a theory of final embeddings like that we have given may aid efforts to understand how they differ from hidden representations.

This work opens new avenues of research from both theoretical and empirical angles. For theory, we draw new connections between SSL, matrix factorization, and questions of inductive bias which admit further study. Empirically, it seems plausible that our a picture of SSL learning can enable algorithmic improvements that give faster, more robust, or better-generalizing training. The prospects for accelerating the training of SSL methods, which typically require many more steps to converge than standard supervised methods, seem particularly promising. For example, our experiments suggest that this slow training may be due to the long times required for lower-eigenvalue modes to emerge, and that an optimizer or loss function that focuses updates on near-zero eigendirections in the embedding space may speed up training without sacrificing stability or generalization. We describe several potential paths for realizing this speedup in Appendix G and encourage practitioners to explore their implementation.

The clear occurrence of stepwise learning far outside the NTK regime suggests it ought to be derivable from a far less restrictive set of assumptions on the model. We leave the development of a more generic theory for future work. A promising starting point may be the view of stepwise learning as a consequence of symmetry-breaking which is discussed in Appendix F.

Another interesting direction is the observation of stepwise behavior in masked-image modeling frameworks, which currently constitute a large fraction of the SSL literature (Baevski et al., 2022; He et al., 2022; Assran et al., 2023). We suspect that, upon small initialization, their hidden representations may also exhibit stepwise dimensionality growth as the system learns to pass more and more information from input to output.

Acknowledgements

The authors thank Bobak Kiani, Randall Balestreiro, Vivien Cabannes, Kanjun Qiu, Ellie Kitanidis, Bryden Fogelman, Bartosz Wróblewski, Nicole Seo, Nikhil Vyas, and Michael DeWeese for useful discussions and comments on the manuscript. JS gratefully acknowledges support from the National Science Foundation Graduate Fellow Research Program (NSF-GRFP) under grant DGE 1752814.

References

Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019.

Arora, S., Cohen, N., and Hazan, E. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pp. 244–253. PMLR, 2018.

Arora, S., Cohen, N., Hu, W., and Luo, Y. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019a.

Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332. PMLR, 2019b.

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019c.

Ash, J. T., Goel, S., Krishnamurthy, A., and Misra, D. Investigating the role of negatives in contrastive representation learning. *arXiv preprint arXiv:2106.09943*, 2021.

Assran, M., Balestriero, R., Duval, Q., Bordes, F., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., and Ballas, N. The hidden uniform cluster prior in self-supervised learning. *arXiv preprint arXiv:2210.07277*, 2022a.

Assran, M., Caron, M., Misra, I., Bojanowski, P., Bordes, F., Vincent, P., Joulin, A., Rabbat, M., and Ballas, N. Masked siamese networks for label-efficient learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pp. 456–473. Springer, 2022b.

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., and Ballas, N. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023.

Atanasov, A., Bordelon, B., and Pehlevan, C. Neural networks as kernel learners: The silent alignment effect. *arXiv preprint arXiv:2111.00034*, 2021.

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., and Auli, M. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, pp. 1298–1312. PMLR, 2022.

Balestriero, R. and LeCun, Y. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *arXiv preprint arXiv:2205.11508*, 2022.

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

Bordelon, B. and Pehlevan, C. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *arXiv preprint arXiv:2205.09653*, 2022.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Cabannes, V., Kiani, B. T., Balestriero, R., LeCun, Y., and Bietti, A. The ssl interplay: Augmentations, inductive bias, and generalization. *arXiv preprint arXiv:2302.02774*, 2023.

Canatar, A., Bordelon, B., and Pehlevan, C. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature communications*, 12(1):1–12, 2021.

Cao, Y., Fang, Z., Wu, Y., Zhou, D.-X., and Gu, Q. Towards understanding the spectral bias of deep learning. *arXiv preprint arXiv:1912.01198*, 2019.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

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, 2020.

Chen, T., Luo, C., and Li, L. Intriguing properties of contrastive losses. *Advances in Neural Information Processing Systems*, 34:11834–11845, 2021.

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.

Chi, Y., Lu, Y. M., and Chen, Y. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269, 2019.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pp. 1675–1685. PMLR, 2019.

Du, S. S., Hu, W., and Lee, J. D. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. *Advances in Neural Information Processing Systems*, 31, 2018.

Fort, S., Dziugaite, G. K., Paul, M., Kharaghani, S., Roy, D. M., and Ganguli, S. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *Advances in Neural Information Processing Systems*, 33: 5850–5861, 2020.

Fukumizu, K. Effect of batch learning in multilayer neural networks. *Gen*, 1(04):1E–03, 1998.

Garrido, Q., Balestriero, R., Najman, L., and Lecun, Y. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. *arXiv preprint arXiv:2210.02885*, 2022.

Geirhos, R., Narayanappa, K., Mitzkus, B., Bethge, M., Wichmann, F. A., and Brendel, W. On the surprising similarities between supervised and self-supervised models. *arXiv preprint arXiv:2010.08377*, 2020.

Grigg, T. G., Busbridge, D., Ramapuram, J., and Webb, R. Do self-supervised and supervised methods learn similar visual representations? *arXiv preprint arXiv:2110.00528*, 2021.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems*, 30, 2017.

HaoChen, J. Z., Wei, C., Gaidon, A., and Ma, T. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021.

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, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Henaff, O. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pp. 4182–4192. PMLR, 2020.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Horace He, R. Z. functorch: Jax-like composable function transforms for pytorch. `https://github.com/pytorch/functorch`, 2021.

Jacot, A., Hongler, C., and Gabriel, F. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Jacot, A., Ged, F., Simsek, B., Hongler, C., and Gabriel, F. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity. *arXiv preprint arXiv:2106.15933*, 2021.

Jacot, A., Golikov, E., Hongler, C., and Gabriel, F. Feature learning in l_2-regularized dnns: Attraction/repulsion and sparsity. *arXiv preprint arXiv:2205.15809*, 2022.

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

Johnson, D. D., Hanchi, A. E., and Maddison, C. J. Contrastive learning can find an optimal basis for approximately view-invariant functions. *arXiv preprint arXiv:2210.01883*, 2022.

Kardar, M. *Statistical physics of fields*. Cambridge University Press, 2007.

Kiani, B. T., Balestriero, R., Chen, Y., Lloyd, S., and LeCun, Y. Joint embedding self-supervised learning in the kernel regime. *arXiv preprint arXiv:2209.14884*, 2022.

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

Landau, L. D. On the problem of turbulence. In *Dokl. Akad. Nauk USSR*, volume 44, pp. 311, 1944.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2018.

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Li, Y., Ma, T., and Zhang, H. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pp. 2–47. PMLR, 2018.

Long, P. M. Properties of the after kernel. *arXiv preprint arXiv:2105.10585*, 2021.

Loureiro, B., Gerbelot, C., Cui, H., Goldt, S., Krzakala, F., Mezard, M., and Zdeborová, L. Learning curves of generic features maps for realistic datasets with a teacher-student model. *Advances in Neural Information Processing Systems*, 34:18137–18151, 2021.

Mallinar, N., Simon, J. B., Abedsoltan, A., Pandit, P., Belkin, M., and Nakkiran, P. Benign, tempered, or catastrophic: A taxonomy of overfitting. *arXiv preprint arXiv:2207.06569*, 2022.

Mei, S. and Montanari, A. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 2019.

Nozawa, K. and Sato, I. Understanding negative samples in instance discriminative self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34:5784–5797, 2021.

Pokle, A., Tian, J., Li, Y., and Risteski, A. Contrasting the landscape of contrastive and non-contrastive learning. *arXiv preprint arXiv:2203.15702*, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Rahimi, A., Recht, B., et al. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 3, pp. 5, 2007.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Roberts, D. A., Yaida, S., and Hanin, B. *Frontmatter*. Cambridge University Press, 2022.

Saunshi, N., Ash, J., Goel, S., Misra, D., Zhang, C., Arora, S., Kakade, S., and Krishnamurthy, A. Understanding contrastive learning requires incorporating inductive biases. *arXiv preprint arXiv:2202.14037*, 2022.

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.

Shawe-Taylor, J., Cristianini, N., et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

Simon, J. B., Dickens, M., Karkada, D., and DeWeese, M. R. The eigenlearning framework: A conservation law perspective on kernel ridge regression and wide neural networks. *arXiv preprint arXiv:2110.03922*, 2021.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33: 7537–7547, 2020.

Tian, Y., Yu, L., Chen, X., and Ganguli, S. Understanding self-supervised learning with dual deep networks. *arXiv preprint arXiv:2010.00578*, 2020.

Tian, Y., Chen, X., and Ganguli, S. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pp. 10268–10278. PMLR, 2021.

Tosh, C., Krishnamurthy, A., and Hsu, D. Contrastive estimation reveals topic posterior information to linear models. *J. Mach. Learn. Res.*, 22:281–1, 2021a.

Tosh, C., Krishnamurthy, A., and Hsu, D. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pp. 1179–1206. PMLR, 2021b.

Tsai, Y.-H. H., Wu, Y., Salakhutdinov, R., and Morency, L.-P. Self-supervised learning from a multi-view perspective. *arXiv preprint arXiv:2006.05576*, 2020.

Tsai, Y.-H. H., Ma, M. Q., Yang, M., Zhao, H., Morency, L.-P., and Salakhutdinov, R. Self-supervised representation learning with relative predictive coding. *arXiv preprint arXiv:2103.11275*, 2021.

Vyas, N., Bansal, Y., and Nakkiran, P. Limitations of the ntk for understanding generalization in deep learning. *arXiv preprint arXiv:2206.10012*, 2022.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Wei, A., Hu, W., and Steinhardt, J. More than a toy: Random matrix models predict how real-world neural representations generalize. In *International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2022.

Wei, C., Shen, K., Chen, Y., and Ma, T. Theoretical analysis of self-training with deep networks on unlabeled data. *arXiv preprint arXiv:2010.03622*, 2020.

Wen, Z. and Li, Y. Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conference on Machine Learning*, pp. 11112–11122. PMLR, 2021.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Yang, G. Tensor programs I: wide feedforward or recurrent neural networks of any architecture are gaussian processes. *CoRR*, abs/1910.12478, 2019. URL `http://arxiv.org/abs/1910.12478`.

Yang, G. and Hu, E. J. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pp. 11727–11737. PMLR, 2021.

Yang, G. and Salman, H. A fine-grained spectral perspective on neural networks. *arXiv preprint arXiv:1907.10599*, 2019.

You, Y., Gitman, I., and Ginsburg, B. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Ziyin, L., Lubana, E. S., Ueda, M., and Tanaka, H. What shapes the loss landscape of self-supervised learning? *arXiv preprint arXiv:2210.00638*, 2022.

A. Additional figures

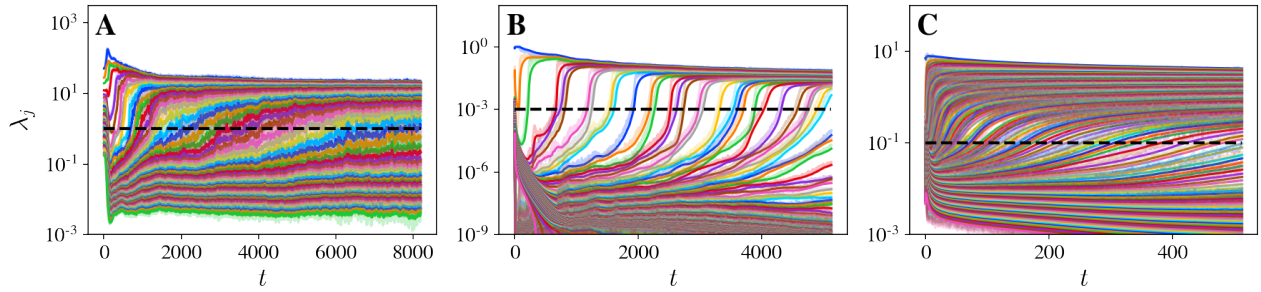

Figure 3: **Stepwise learning in SSL with ResNets with standard initialization and hyperparameters.** Embedding eigenvalues over time for **(A)** Barlow Twins, **(B)** SimCLR, and **(C)** VICReg. Dashed horizontal lines show rough eigenvalue thresholds separating the cluster of modes which have grown from the cluster of modes which have not yet grown and match the thresholds of Figure 4.

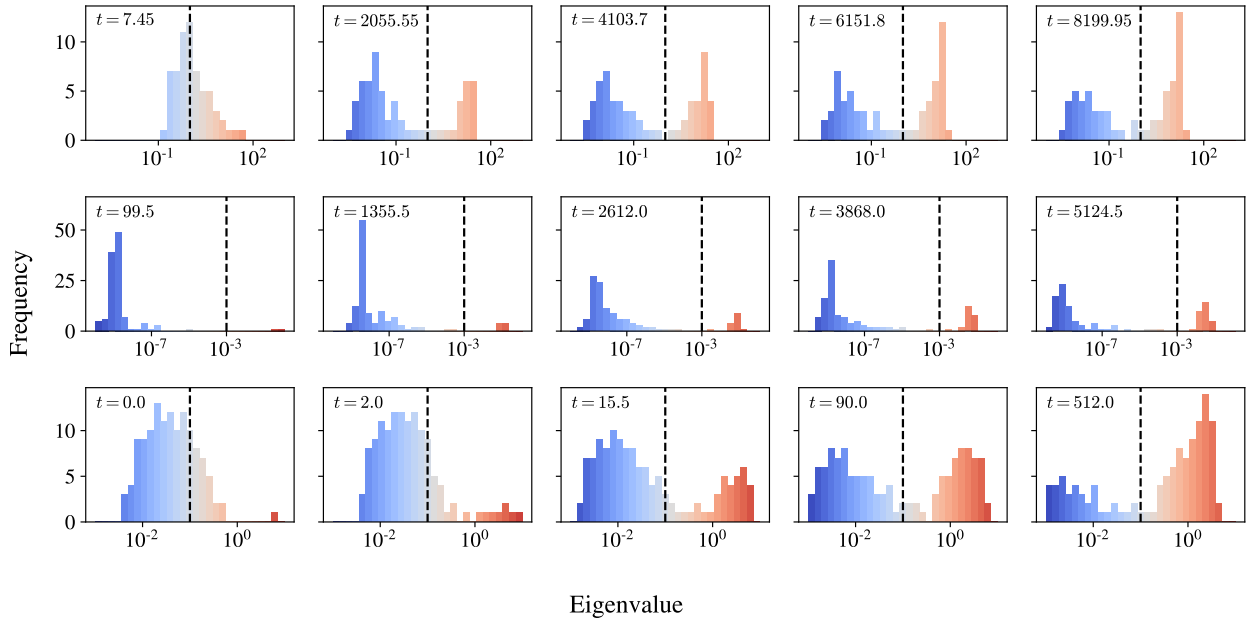

Figure 4: **Bimodal distribution of embedding eigenvalues shown over time for SSL with ResNets with standard initialization and hyperparameters.**. Histograms of embedding eigenvalues at selected times throughout training for **Top row:** Barlow Twins. **Middle row:** SimCLR. **Bottom row:** VICReg. Dashed vertical lines indicate the same eigenvalue thresholds as in Figure 3.

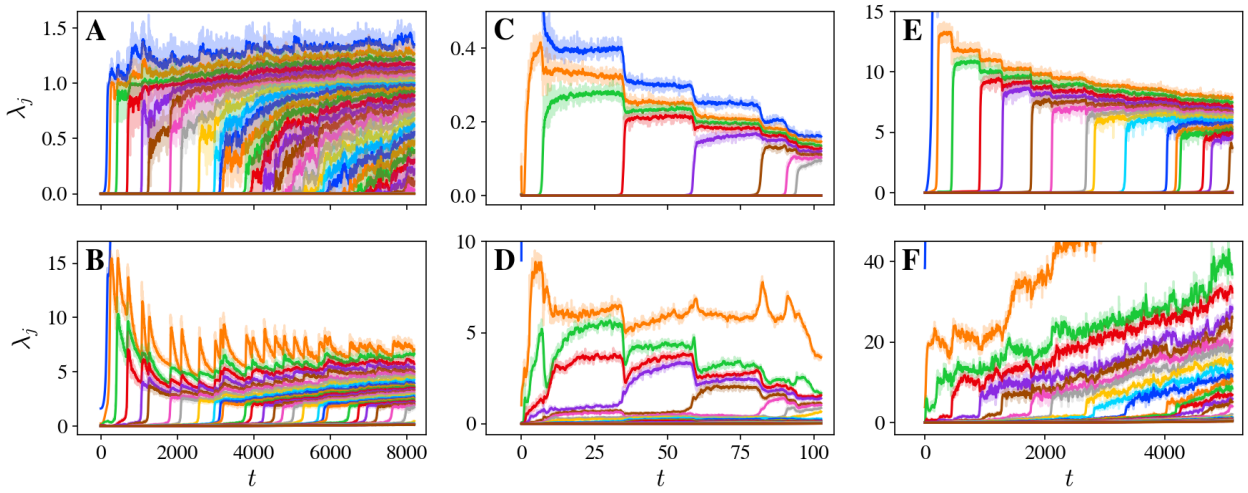

Figure 5: **Hidden representations exhibit learning steps aligned with those of embeddings.** Eigenvalues of embeddings (top row) and hidden representations (bottom row) vs time $t = [\text{lr}] \times [\text{step}]$ for **(A, B)** a ResNet trained with Barlow Twins loss, **(C, D)** a ResNet trained with SimCLR loss, and **(E, F)** a ResNet trained with VICReg loss, all trained on STL-10 with small initialization.

B. Proofs

B.1. Proof of Proposition 4.3

We wish to show that the top spectral $\boldsymbol{W}$ are the unique solutions to

$$\underset{\boldsymbol{W}}{\text{argmin}} \|\boldsymbol{W}\|_F \quad \text{s.t.} \quad \mathcal{L}(\boldsymbol{W}) = 0. \tag{26}$$

Any such solution is also a minimum of

$$\mathcal{L}_\epsilon(\boldsymbol{W}) \equiv \mathcal{L}(\boldsymbol{W}) + \|\boldsymbol{W}\|_F^2 \tag{27}$$

as $\epsilon \to 0$. We will show that the set of minima are precisely the top spectral embeddings.

Any local minimum must satisfy

$$\nabla_{\boldsymbol{W}} \mathcal{L}_\epsilon = (\boldsymbol{I} - \boldsymbol{W}\boldsymbol{\Gamma}\boldsymbol{W}^\top)\boldsymbol{W}\boldsymbol{\Gamma} - \epsilon \boldsymbol{W} = \boldsymbol{0} \tag{28}$$

$$\Rightarrow \boldsymbol{W}^\top \nabla_{\boldsymbol{W}} \mathcal{L}_\epsilon = \boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{\Gamma} - \left(\boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{\Gamma}\right)^2 - \epsilon \boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{0}. \tag{29}$$

Note that the first and second terms of Equation 29 share all eigenvectors (i.e. commute), and thus the third term must commute with both of them. From the fact that $\boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{\Gamma}$ and $\boldsymbol{W}^\top \boldsymbol{W}$ share eigenvectors, we can conclude that $\boldsymbol{W}^\top \boldsymbol{W}$ and $\boldsymbol{\Gamma}$ share eigenvectors. The eigenvectors of $\boldsymbol{W}^\top \boldsymbol{W}$ are simply the right singular vectors of $\boldsymbol{W}$.

Any local minimum must thus take the form

$$\boldsymbol{W} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{\Gamma}^{(\mathcal{J})}, \tag{30}$$

where

- $\boldsymbol{U} \in \mathbb{R}^{d \times d}$ is orthogonal,
- $\boldsymbol{S} \in \mathbb{R}^{d \times d} = \text{diag}(s_1, ..., s_d)$,
- $\mathcal{J}$ is a list of d elements from $\{1, ..., d\}$,
- $\boldsymbol{\Gamma}^{(\mathcal{J})} \in \mathbb{R}^{d \times m}$ contains as rows the d eigenvalues of $\boldsymbol{\Gamma}$ corresponding to the elements of $\mathcal{J}$, and

- the singular values satisfy

$$s_j^2\gamma_{\mathcal{J}_j} - s_j^4\gamma_{\mathcal{J}_j}^2 - \epsilon s_j^2 = 0 \quad \Rightarrow \quad s_j \in \left\{0, \sqrt{\frac{\gamma_{\mathcal{J}_j} - \epsilon}{\gamma_{\mathcal{J}_j}^2}}\right\}, \tag{31}$$

  where $s_j = 0$ is chosen by default when $\gamma_{\mathcal{J}_j} \leq 0$.

Of these candidate local minima, a global minimum must choose $s_j = 0$ as infrequently as possible, and given that must minimize $\|\boldsymbol{W}\|_F^2 = \sum_j s_j^2$. This is achieved by choosing $\mathcal{J} = (1, ..., d)$, which yields the top spectral solutions.

B.2. Proof of Proposition 5.1

We want to translate our results for linear models in Section 4 to the kernel setting by phrasing interesting quantities in terms of inner products $\boldsymbol{x}_i^\top \boldsymbol{x}_j$. After finishing this procedure, we will reinterpret $\boldsymbol{x}_i^\top \boldsymbol{x}_j \to \boldsymbol{\phi}(x_i)^\top \boldsymbol{\phi}(x_j) = \mathcal{K}(x_i, x_j)$.

It will be useful to perform some setup before diving in. We define $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_d]^\top \in \mathbb{R}^{n\times m}$ and $\boldsymbol{X}' = [\boldsymbol{x}'_1, ..., \boldsymbol{x}'_d]^\top$ and define the full dataset kernel matrix $\tilde{\boldsymbol{K}} = \begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{X}'\end{bmatrix}\begin{bmatrix}\boldsymbol{X}^\top & \boldsymbol{X}'^\top\end{bmatrix}$. Recall that $\boldsymbol{\Gamma} = \frac{1}{2n}\left(\boldsymbol{X}^\top\boldsymbol{X}' + \boldsymbol{X}'^\top\boldsymbol{X}\right)$. In what follows, all interesting vectors will lie in either the row-space or column-space of $\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{X}'\end{bmatrix}$ (whichever is appropriate), and so all vectors over the dataset (i.e. with $2n$ elements) will lie in the cokernel of $\tilde{\boldsymbol{K}}$, so we are free to use the pseudoinverse of $\tilde{\boldsymbol{K}}$ without worrying about the action of null eigendirections. All matrix inverses henceforth will be interpreted as pseudoinverses.

Proof of clause (a). An eigenpair $(\gamma, \boldsymbol{g})$ of $\boldsymbol{\Gamma}$ satisfies

$$\boldsymbol{\Gamma}\boldsymbol{g} = \gamma\boldsymbol{g}. \tag{32}$$

Assume that $\gamma \neq 0$ and $\|\boldsymbol{g}\| = 1$. Let us define a vector $\boldsymbol{b}$ dual to $\boldsymbol{g}$ as

$$\boldsymbol{b} = \tilde{\boldsymbol{K}}^{-1/2}\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{X}'\end{bmatrix}\boldsymbol{g} \quad \Leftrightarrow \quad \boldsymbol{g} = \begin{bmatrix}\boldsymbol{X}^\top & \boldsymbol{X}'^\top\end{bmatrix}\tilde{\boldsymbol{K}}^{-1/2}\boldsymbol{b}. \tag{33}$$

Note that $\|\boldsymbol{g}\| = \|\boldsymbol{b}\|$. Plugging Equation 33 into Equation 32, we find that

$$\boldsymbol{\Gamma}\begin{bmatrix}\boldsymbol{X}^\top & \boldsymbol{X}'^\top\end{bmatrix}\tilde{\boldsymbol{K}}^{-1/2}\boldsymbol{b} = \gamma\begin{bmatrix}\boldsymbol{X}^\top & \boldsymbol{X}'^\top\end{bmatrix}\tilde{\boldsymbol{K}}^{-1/2}\boldsymbol{b}. \tag{34}$$

Left-multiplying both sides by $\tilde{\boldsymbol{K}}^{-1/2}\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{X}'\end{bmatrix}$ yields

$$\tilde{\boldsymbol{K}}^{-1/2}\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{X}'\end{bmatrix}\boldsymbol{\Gamma}\begin{bmatrix}\boldsymbol{X}^\top & \boldsymbol{X}'^\top\end{bmatrix}\tilde{\boldsymbol{K}}^{-1/2}\boldsymbol{b} = \gamma\tilde{\boldsymbol{K}}^{-1/2}\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{X}'\end{bmatrix}\begin{bmatrix}\boldsymbol{X}^\top & \boldsymbol{X}'^\top\end{bmatrix}\tilde{\boldsymbol{K}}^{-1/2}\boldsymbol{b} \tag{35}$$

$$= \gamma\boldsymbol{b}. \tag{36}$$

Defining

$$\boldsymbol{Z} \equiv \begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{X}'\end{bmatrix}\boldsymbol{\Gamma}\begin{bmatrix}\boldsymbol{X}^\top & \boldsymbol{X}'^\top\end{bmatrix} = \frac{1}{2n}\begin{bmatrix}\boldsymbol{X}\boldsymbol{X}'^\top\boldsymbol{X}\boldsymbol{X}^\top & \boldsymbol{X}\boldsymbol{X}'^\top\boldsymbol{X}\boldsymbol{X}'^\top\\ \boldsymbol{X}'\boldsymbol{X}'^\top\boldsymbol{X}\boldsymbol{X}^\top & \boldsymbol{X}'\boldsymbol{X}'^\top\boldsymbol{X}\boldsymbol{X}'^\top\end{bmatrix} + \frac{1}{2n}\begin{bmatrix}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{X}'\boldsymbol{X}^\top & \boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{X}'\boldsymbol{X}'^\top\\ \boldsymbol{X}'\boldsymbol{X}^\top\boldsymbol{X}'\boldsymbol{X}^\top & \boldsymbol{X}'\boldsymbol{X}^\top\boldsymbol{X}'\boldsymbol{X}'^\top\end{bmatrix} \tag{37}$$

as in the main text and $\boldsymbol{K}_{\boldsymbol{\Gamma}} \equiv \tilde{\boldsymbol{K}}^{-1/2}\boldsymbol{Z}\tilde{\boldsymbol{K}}^{-1/2}$, we find that γ is also an eigenvalue of $\boldsymbol{K}_{\boldsymbol{\Gamma}}$. The same argument run in reverse (now rewriting $\boldsymbol{b}$ in terms of $\boldsymbol{g}$) yields that nonzero eigenvalues of $\tilde{\boldsymbol{K}}_{\boldsymbol{\Gamma}}$ are also eigenvalues of $\boldsymbol{\Gamma}$. □

Proof of clause (b). The top spectral weights given by Definition 4.2 yield the spectral representation

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{U}\tilde{\boldsymbol{S}}[\boldsymbol{g}_1 \ ... \ \boldsymbol{g}_d]^\top\boldsymbol{x}. \tag{38}$$

Plugging in Equation 33 yields that

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{U}\tilde{\boldsymbol{S}}[\boldsymbol{b}_1 \ ... \ \boldsymbol{b}_d]^\top\tilde{\boldsymbol{K}}^{-1/2}\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{X}'\end{bmatrix}\boldsymbol{x} \tag{39}$$

as claimed by Proposition 5.1. □

Proof of clause (c). The RKHS norm of a function with respect to a kernel $\mathcal{K}$ is equivalent to the minimal Frobenius norm of an explicit matrix transformation on the hidden features of the kernel, so this clause follows from Proposition 4.3. □

Kernelized dynamics.

Via direct kernelization of Result 4.6, the kernelized representations trained from small init at an arbitrary finite time t are well-approximated by

$$\boldsymbol{f}(\boldsymbol{x}, t) = \boldsymbol{U}\tilde{\boldsymbol{S}}(t)[\boldsymbol{g}_1 \dots \boldsymbol{g}_d]^\top \boldsymbol{x} = \boldsymbol{U}\tilde{\boldsymbol{S}}(t)[\boldsymbol{b}_1 \dots \boldsymbol{b}_d]^\top \tilde{\boldsymbol{K}}^{-1/2}[\boldsymbol{K}_{x\mathcal{X}}\ \boldsymbol{K}_{x\mathcal{X}'}]^\top, \tag{40}$$

with the singular values on the diagonal of $\tilde{\boldsymbol{S}}(t)$ evolving according to Proposition 4.1. The initialization-dependent matrix $\boldsymbol{U}$ and effective initial singular values are found in the explicit linear case as

$$\boldsymbol{U}\tilde{\boldsymbol{S}}(0) = \mathcal{A}(\boldsymbol{W}(0){\boldsymbol{\Gamma}^{(\leq d)}}^\top) = \mathcal{A}(\boldsymbol{W}(0)[\boldsymbol{g}_1, \dots, \boldsymbol{g}_d]). \tag{41}$$

Using Equation 33, we then have that

$$\boldsymbol{U}\tilde{\boldsymbol{S}}(0) = \mathcal{A}\left([\boldsymbol{f}(x_1), \dots, \boldsymbol{f}(x_n), \boldsymbol{f}(x_1'), \dots, \boldsymbol{f}(x_n')]\tilde{\boldsymbol{K}}^{-1/2}[\boldsymbol{b}_1, \dots \boldsymbol{b}_d]\right). \tag{42}$$

Equations 40 and 42 together permit one to find the trajectories from arbitrary small init in fully kernelized form.

C. Derivation of dynamics from generic small initialization

Here we give an informal derivation of Result 4.6, which states that the solution of Equation 6 from arbitrary initialization with scale $\alpha \ll 1$ closely matches the analytical solution from a certain spectrally aligned initialization. Recall that our ingredients are the following:

- $\gamma_1, \dots, \gamma_d$ are unique and positive.
- $\tilde{\boldsymbol{W}}_0 \in \mathbb{R}^{d \times m}$ with $\tilde{\boldsymbol{W}}_0 \boldsymbol{\Gamma}^{(\leq d)}$ full rank.
- $\boldsymbol{W}(t)$ is the true solution with $\boldsymbol{W}(0) = \alpha \tilde{\boldsymbol{W}}_0$.
- $\boldsymbol{W}^*(t)$ is the spectrally aligned solution with $\boldsymbol{W}^*(0) = \mathcal{A}(\alpha \tilde{\boldsymbol{W}}_0)$ whose dynamics are given exactly by Proposition 4.1.

We will show that, for sufficiently small α, the true and aligned solutions remain arbitrarily close.

We will find it convenient to parameterize $\boldsymbol{W}(t)$ as

$$\boldsymbol{W}(t) = \boldsymbol{U} \begin{bmatrix} s_1(t) & a_{12}(t) & a_{13}(t) & \cdots & a_{1d}(t) & \cdots & a_{1m}(t) \\ a_{21}(t) & s_2(t) & a_{23}(t) & \cdots & a_{2d}(t) & \cdots & a_{2m}(t) \\ a_{31}(t) & a_{32}(t) & s_3(t) & \cdots & a_{3d}(t) & \cdots & a_{3m}(t) \\ \vdots & \vdots & \vdots & \ddots & \vdots & & \vdots \\ a_{d1}(t) & a_{d2}(t) & a_{d3}(t) & \cdots & s_d(t) & \cdots & a_{dm}(t) \end{bmatrix} \boldsymbol{\Gamma}^{(\leq m)} \tag{43}$$

$$= \boldsymbol{U}\boldsymbol{A}(t)\boldsymbol{\Gamma}^{(\leq m)}, \tag{44}$$

where $s_j(0) > 0$ and $a_{jk}(0) = 0$ for all $j > k$. We use the special notation s_j for the diagonal elements of $\boldsymbol{A}$ to foreshadow that these will act as the effective singular values of the dynamics. Note that the spectrally aligned initialization $\mathcal{A}(\tilde{\boldsymbol{W}}_0)$ is precisely the $\boldsymbol{W}(0)$ of Equation 43 but with all off-diagonal entries of $\boldsymbol{A}(0)$ zeroed out. Our strategy will be to show that no $a_{jk}(t)$ ever grows sufficiently large to affect the dynamics to leading order, and thus $\boldsymbol{W}(t)$ and $\boldsymbol{W}^*(t)$ remain close.

We will make use of big-$\mathcal{O}$ notation to describe the scaling of certain values with α. Eigenvalues γ_j and differences $\gamma_j - \gamma_{j+1}$ will be treated as constants. Note that, because $\boldsymbol{W}(0) = \alpha \tilde{\boldsymbol{W}}_0$, all elements of $\boldsymbol{A}(0)$ are $\Theta(\alpha)$ if they are not zero.

Diagonalization of dynamics. Define $\mathbf{\Lambda} = \text{diag}(\gamma_1, ..., \gamma_m)$. The dynamics of Equation 6 state that $\boldsymbol{A}(t)$ evolves as

$$\frac{d\boldsymbol{A}(t)}{dt} = \left(\boldsymbol{I} - \boldsymbol{A}(t)\mathbf{\Lambda}\boldsymbol{A}^\top(t)\right)\boldsymbol{A}(t)\mathbf{\Lambda}, \tag{45}$$

where we have reparameterized $t \to t/4$ to absorb the superfluous prefactor of 4.

Approximate solution to dynamics. So long as all a_{jk} remain small (i.e. $o(1)$), then these dynamics are given by

$$\frac{d\boldsymbol{A}(t)}{dt} \approx \left(\boldsymbol{I} - \begin{bmatrix} \gamma_1 s_1^2(t) & & & & \\ & \gamma_2 s_2^2(t) & & & \\ & & \gamma_3 s_3^2(t) & & \\ & & & \ddots & \\ & & & & \gamma_d s_d^2(t) \end{bmatrix}\right)\boldsymbol{A}(t)\mathbf{\Lambda}. \tag{46}$$

We will show that all a_{jk} indeed remain small under the evolution of Equation 46, and so Equation 46 remains valid.

Solving Equation 46 yields

$$s_j(t) = \frac{e^{\gamma_j t}}{\sqrt{s_j^{-2}(0) + (e^{2\gamma_j t} - 1)\gamma_j}} \tag{47}$$

$$a_{jk}(t) = \begin{cases} a_{jk}(0)\left(\frac{s_j(t)}{s_j(0)}\right)^{\gamma_k/\gamma_j} = \mathcal{O}\left(\alpha^{1-\gamma_k/\gamma_j}\right) & \text{for } j < k, \\ 0 & \text{for } j > k. \end{cases} \tag{48}$$

As discussed in the main text, each $s_j(t)$ remains small up to a time $\tau_j \sim -\gamma_j^{-1}\log\alpha$, at which it quickly grows until $\gamma_j s_j^2(t) = 1$ and saturates. Entries of $\boldsymbol{A}(t)$ below the diagonal remain zero. Entries of $\boldsymbol{A}(t)$ above the diagonal exhibit interesting dynamics: entry $a_{jk}(t)$ with $j < k$ grows exponentially at rate γ_k, but its growth is curtailed by the saturation of $s_j(t)$ before it has time to reach order one. This is because each $s_j(t)$ grows faster than all $a_{jk}(t)$ in its row.

All $a_{jk}(t)$ thus remain $o(1)$ and all $s_j(t)$ closely follow the ideal solution of Equation 10, and so $\|\boldsymbol{W}(t) - \boldsymbol{W}^*(t)\|_F$ remains $o(1)$. This concludes the derivation.

Numerical validation of approximation. While the numerical experiment presented in Figure 1 validates our claim regarding the trajectory of $\boldsymbol{W}(t)$ from generic small initialization closely matching theoretical predictions from aligned initialization, here we go a step further and show agreement for individual elements of $\boldsymbol{A}(t)$. We analytically solve Equation 46 for $d = 3$ and $m = 5$ with $\gamma_j = 2^{-j}$, starting from a random upper-triangular $\boldsymbol{A}(0)$ of scale $\alpha = 10^{-9}$. The results, plotted in Figure 6, closely match the unapproximated dynamics of Equations 47 and 48.

D. Experimental details

Below we describe our suite of SSL experiments. All code and experiment configs are available at `https://gitlab.com/generally-intelligent/ssl_dynamics`.

D.1. Single-hidden-layer MLP

This experiment follows the linear model numerical experiment described in Section 4.4. We train a single-hidden-layer MLP for 7000 epochs over a fixed batch of 50 images from CIFAR10 using full-batch SGD. Each image is subject to a random 20x20 crop and no other augmentations. The learning rate is $\eta = 0.0001$ and weights are scaled upon initialization by $\alpha = 0.0001$. The hidden layer has width 2048 and the network output dimension is $d = 10$. We use Barlow Twins loss, but do not apply batch norm to the embeddings when calculating the cross-correlation matrix. λ is set to 1.

D.2. Full-size models

We run two sets of experiments experiments with ResNet-50 encoders which respectively use *standard init* and *small init* whose results appear in figures in this paper. The standard set aims to reproduce performance reported in source publications and match original hyperparameters, whereas the small-init set includes modifications which aim to bring out the stepwise

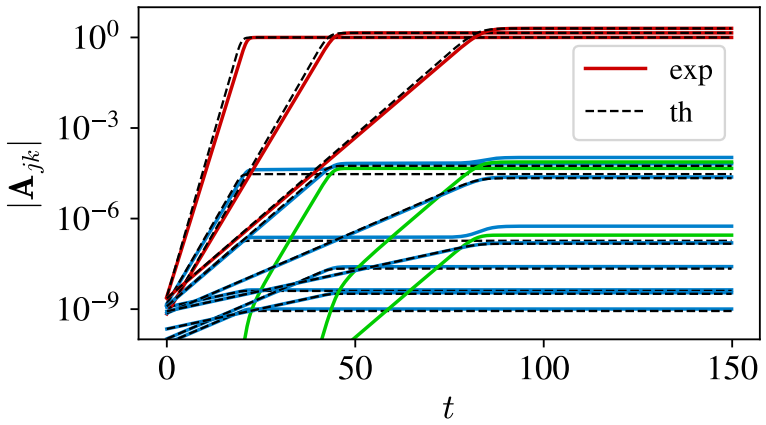

Figure 6: True $|s_j(t)|$ and $|a_{jk}(t)|$ compared with theoretical predictions from small init. Red traces show $s_j(t)$, blue traces show $a_{jk}(t)$ with $j < k$, and green traces show a_{jk} with $j > k$. While there are no theoretical traces for a_{jk} with $j > k$, these elements do remain small as predicted.

learning dynamics predicted by our theory more clearly. Relative to the standard set, the small init set multiplies all initial weights by a small scale factor α and uses a reduced learning rate η. The parameters used for each method are generally identical across the two sets except for α and η.

For all experiments in the standard set, we keep parameters as close as possible to those in the source publication but make minor adjustments where we find they improve evaluation metrics. Additionally, in order to avoid a steep compute budget for reproduction, we train our models on STL-10 instead of ImageNet, and reduce batch size, projector layer widths and training epochs so that the models can be trained using a single GPU in under 24 hours. **Augmentations, optimizer parameters such as momentum and weight decay, and learning rate schedules are generally left unchanged from the source publications.** Below we describe deviations from the stock hyperparameters.

Barlow Twins. We set batch size to 64, and update the projector to use single hidden layer of width 2048 and embedding size $d = 512$. We use the LARS optimizer (You et al., 2017) with stock learning rates. In the small-init case, we additionally set $\lambda = 1$ and $\alpha = 0.093$ and remove the pre-loss batch norm. In the standard case, we keep $\lambda = 0.0051$ as suggested by Zbontar et al. We train the networks for 320 epochs.

SimCLR. We set batch size to 256, the cross-entropy temperature $\tau = 0.2$ and $\eta = 0.5$. We train using SGD with weight decay $1 * 10^{-5}$ and no LR scheduling. In the small-init case, we use $\eta = 0.01$ and additionally set $\alpha = 0.087$. We train the networks for 320 epochs.[18]

VICReg. We set batch size to 256, reduce the projector's hidden layer width to 512 and set $d = 128$. We use $\eta = 0.5$ and weight decay $1 * 10^{-5}$ and reduce the warmup epochs to 1. We use the LARS optimizer. In the small-init case, we use $lr = 0.25$ and additionally set $\alpha = 0.133$. We train the networks for 500 epochs.

Values of α. The init scales α quoted above were tuned so as to give clear stepwise behavior while also not slowing training too much. We found that a useful heuristic is to choose α such that the top eigenvalue at init is roughly 10^{-3} for Barlow Twins and VICReg and 10^{-6} for SimCLR.

D.3. Evaluating performance

Our small init experiments cleanly display stepwise learning. Here we show that these hyperparameter changes do not dramatically worsen model performance. We evaluate performance by measuring the quality of learned hidden representations using a linear probe, and perform this evaluation many times throughout training. Validation accuracies

[18]We found that SimCLR showed noticeably more apparent stepwise behavior in the *standard* case than did Barlow Twins or VICReg (see Figure 3), with steps visible even in the loss curve (not shown). We are rather surprised by this difference and invite attempts to replicate it.

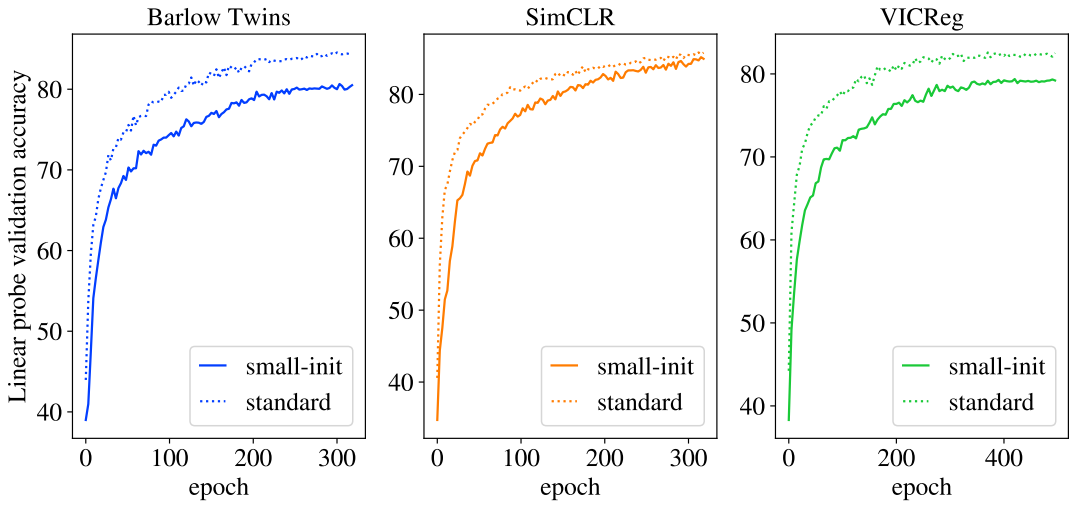

Figure 7: Validation accuracy of a linear probe throughout training for our standard and small-init experiments.

increase throughout training (Figure 7) as the learned representations improve over time, and generally small-init experiments fare similarly to the standard set, with a small performance drop of a few percent, but train slower as expected.

D.4. Measuring eigenvalues

In the main text, all eigenvalues reported for *embeddings* using the *Barlow Twins loss* are the eigenvalues $\lambda_j(\boldsymbol{C})$, with $\boldsymbol{C}$ the cross-correlation matrix across positive pairs as defined in Section 3. Eigenvalues reported for the embeddings of other losses are the eigenvalues $\lambda(\tilde{\boldsymbol{C}})$ of the (centered) covariance matrix, with $\tilde{\boldsymbol{C}} \equiv \mathbb{E}_x[(\boldsymbol{f}(x) - \bar{\boldsymbol{f}})(\boldsymbol{f}(x) - \bar{\boldsymbol{f}})^\top]$ and $\bar{\boldsymbol{f}} \equiv \mathbb{E}_x[\boldsymbol{f}(x)]$. The eigenvalues reported in Figure 5 for hidden representations are all those of the covariance matrix (but of course computed on the hidden representations instead of the embeddings $\boldsymbol{f}$). We vary the matrix studied because our theory deals explicitly with $\boldsymbol{C}$, but this cross-correlation matrix is not necessarily meaningful except in the embeddings of Barlow Twins. In practice, however, we see similar results for either case (and in fact sometimes even see sharper learning steps in $\lambda_j(\tilde{\boldsymbol{C}})$ than $\lambda_j(\boldsymbol{C})$ even for Barlow Twins).

D.5. Measuring the empirical NTK

In order to calculate the empirical NTK (eNTK), we employ functorch (Horace He, 2021) and express the eNTK as a Jacobian contraction. We find that calculating the full kernel for realistic-sized networks is unfeasible due to memory and runtime constraints, primarily because of the large output dimension d (which is 50 in the experiments of Appendix E) as well as large batch sizes. To overcome this, we employ two tricks. First, we reduce the model output to a scalar by taking an average of the outputs and calculate the eNTK for this modified function instead.[19][20] Second, we chunk up the computation for the full batch into minibatches depending on the size of d and available GPU memory. With this approach, computing the eNTK on 1000 image pairs takes on the order of 1 minute on a single consumer GPU.

E. Predictions of embeddings from the NTK after training

Here we demonstrate that the true embeddings learned by SSL methods using small initialization show fair agreement with predictions computed from our theory using the empirical NTK after training. The motivation for this experiment comes from an observation of Atanasov et al. (2021) regarding ResNets with small initialization trained in a supervised setting. Though the NTK of such a model evolves dramatically during training, the final function is nonetheless well-predicted by kernel regression using the empirical NTK after training. We find a similar result here.

[19] More precisely, we take the average of the d outputs and multiply it by $\sqrt{d}$, a scaling which recovers the original NTK matrix in the ideal, infinite-width case.

[20] Note that a simplification like this is necessary to use our theory: our theory assumes the NTK on n samples is captured by a single matrix, whereas in reality it is a *rank-four tensor* for a model with vector output.

	BT	SimCLR	VICReg	(random subspaces)
$\mathsf{a}(\boldsymbol{P}_{\text{exp}}, \boldsymbol{P}_{\text{th}})$	0.615	0.517	0.592	0.025
$\mathsf{a}(\boldsymbol{P}_{\text{exp}}, \boldsymbol{P}_{\text{NTK}})$	0.510	0.450	0.481	0.025

Table 1: Alignments between true embedding subspaces and those predicted from the final NTK for different SSL methods.

$\mathsf{a}(\boldsymbol{P}^{\text{BT}}_{\text{exp}}, \boldsymbol{P}^{\text{SC}}_{\text{exp}})$	$\mathsf{a}(\boldsymbol{P}^{\text{BT}}_{\text{exp}}, \boldsymbol{P}^{\text{VR}}_{\text{exp}})$	$\mathsf{a}(\boldsymbol{P}^{\text{SC}}_{\text{exp}}, \boldsymbol{P}^{\text{VR}}_{\text{exp}})$	(random subspaces)
0.504	0.504	0.405	0.025

Table 2: Alignments between true embedding subspaces for different SSL methods.

We train ResNet-50 encoders from moderately small init to close to convergence with Barlow Twins ($\alpha = 0.542$), SimCLR ($\alpha = 0.284$), and VICReg ($\alpha = 0.604$) losses on STL-10. These models have only $d = 50$, which is large enough to be nontrivial but small enough to give good statistical power to our subsequent analysis. After training, we then take a batch of $n = 1000$ random augmented image pairs $\mathcal{X}$ and compute both their joint empirical NTK $\tilde{\boldsymbol{K}} \in \mathbb{R}^{2n\times 2n}$ and their embeddings $\boldsymbol{f}_{\text{exp}}(\mathcal{X}) \in \mathbb{R}^{d\times 2n}$. We trust that n is sufficiently larger than d that it is reasonable to treat $\mathcal{X}$ as the full population distribution. We then compute the theoretically predicted embeddings using Equation 24 as

$$\boldsymbol{f}_{\text{th}}(\mathcal{X}) = \tilde{\boldsymbol{S}}[\boldsymbol{b}_1 \dots \boldsymbol{b}_d]^\top \tilde{\boldsymbol{K}}^{1/2}. \tag{49}$$

It is principally the subspace (in function space) spanned by a vector representation which determines the performance of a linear probe on a downstream task. As such, we compute the right singular vectors of $\boldsymbol{f}_{\text{exp}}(\mathcal{X})$ and $\boldsymbol{f}_{\text{th}}$, which we call $\boldsymbol{P}_{\text{exp}} \in \mathbb{R}^{d\times 2n}$ and $\boldsymbol{P}_{\text{th}} \in \mathbb{R}^{d\times 2n}$ and which both have rank d. We then compute the *normalized subspace alignment* $\mathsf{a}(\boldsymbol{P}_{\text{exp}}, \boldsymbol{P}_{\text{th}})$, where $\mathsf{a}(\boldsymbol{P}, \boldsymbol{P}') \equiv \frac{1}{d}\left\|\boldsymbol{P}(\boldsymbol{P}')^\top\right\|_F^2$. This alignment metric attains its maximal value of d when both subspaces contain the same vectors, and has an expectation of $\frac{d^2}{2n} \ll 1$ for random subspaces.[21] As an additional test, we repeated this comparison replacing $\boldsymbol{P}_{\text{th}}$ with $\boldsymbol{P}_{\text{NTK}}$ containing the top d eigenvectors of $\tilde{\boldsymbol{K}}$ and found similar alignment.

We report our observed subspace alignments in Table 1. For all three methods, we see an alignment between 0.5 and 0.6, which is significantly higher than expected by chance, but still misses roughly half of the span of the true embeddings[22]. We anticipate that much of the gap from unity is due to approximation error caused by taking a finite dataset of size $2n$. It is perhaps surprising that our theory seems to work equally well for SimCLR and VICReg as it does for Barlow Twins.

We also report alignment scores *between $\boldsymbol{P}_{exp}$ from the three SSL methods* in Table 2, again finding alignment roughly between 0.4 and 0.5. This finding, independent of any theoretical result, is evidence that these three methods are in some respects learning very similar things.

F. Connection to spontaneous symmetry breaking in physical systems

In the main text, we noted that Landau (1944) encountered essentially the same dynamics in the study of turbulent fluid flow as we found for SSL. This is due to the fact that both are simple processes of *spontaneous symmetry breaking* (SSB), a phenomenon in which a system whose dynamics obey a symmetry spontaneously settles into one of several asymmetric rest states chosen as a result of the system's initial conditions. In this appendix, we will explain how the dynamics of our model of SSL can be understood as a process of SSB.

Recall from Equation 14 that the loss of our toy model upon aligned initialization is

$$\mathcal{L} = \sum_j \mathcal{L}_j = \sum_j (1 - \gamma_j s_j^2)^2 = \sum_j (1 - \bar{s}_j^2)^2, \tag{50}$$

where we define $\mathcal{L}_j = (1 - \gamma_j s_j^2)^2$ and $\bar{s}_j = \gamma_j^{1/2} s_j$. The quantity $\bar{s}_j$ is a (rescaled) singular value of $\boldsymbol{W}$. Singular values are canonically taken to be nonnegative (because one can always enforce this condition by appropriate choice of the singular

[21] One may intuitively think of $\mathsf{a}(\boldsymbol{P}, \boldsymbol{P}')$ as the mean fraction of a random vector from the rowspace of $\boldsymbol{P}$ which is captured by the rowspace of $\boldsymbol{P}'$.

[22] Agreement with predictions from the *initial* NTK (not reported) is generally worse but still greater than chance.

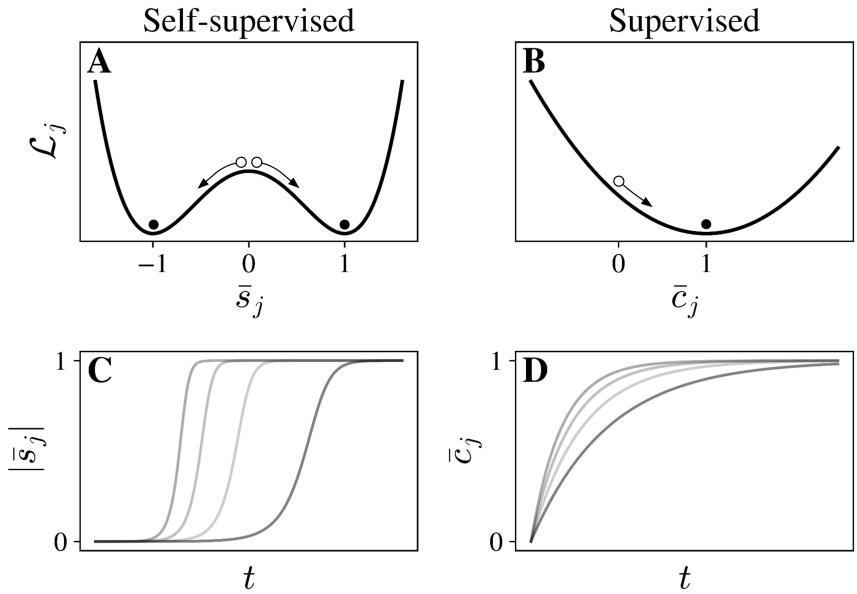

Figure 8: **SSL is a process of symmetry breaking, whereas standard supervised learning is not. (A)**: Nondimensionalized loss landscape for singular value $\bar{s}_j$ in our linear model of Barlow Twins. The system can reach one of two minima depending on the sign of the initialization. **(B)**: Nondimensionalized loss landscape for an eigencoefficient in linearized *supervised* learning with $\bar{c_j}^* = 1$. There is only one local minimum. **(C)**: Example trajectories for the loss landscape in (A) from small init with different eigenvalues γ_j. **(D)**: Example trajectories for the loss landscape in (B) from small init with different eigenvalues γ_j.

vectors), but let us pretend for the present discussion that singular values may take any value along the real line. Note that each singular value evolves via gradient descent according to its respective loss as

$$\bar{s}_j'(t) = \frac{d\mathcal{L}_j}{d\bar{s}_j(t)}. \tag{51}$$

Figure 8A depicts $\mathcal{L}_j$, with trajectories for various γ_j shown in Figure 8C. Note that this 1D landscape is symmetric and has two global minima at $\bar{s}_j = \pm 1$ and one local maximum at $\bar{s}_j = 0$. When the system is initialized near zero, an unstable equilibrium of the dynamics, it will flow towards one basin or the other depending on its initial sign. However, for small $\bar{s}_j(0)$, it takes a time $\tau_j = -\log(|\bar{s}_j(0)|)/4\gamma_j$ for $\bar{s}_j$ to escape the origin,[23] whereas the system thereafter approaches the nearest minimum with the faster timescale $\tau_j' = 1/4\gamma_j$. This slow escape from the origin is what leads to the sharp stepwise behavior we find.

It should be noted that the quartic form of $\mathcal{L}_j$ appears in toy models of SSB across physics (see e.g. Landau-Ginzburg theory (Kardar, 2007)) and is in fact the aforementioned model of Landau (1944). In these and other typical cases, SSB entails the breaking of a symmetry apparent at the outset of the problem such as invariance to translation, rotation, or inversion symmetry. In the case of SSL, the symmetry is the transformation $\boldsymbol{f} \to -\boldsymbol{f}$, which does not change the value of the global loss.[24]

Why does standard supervised learning in the NTK limit not exhibit stepwise behavior upon small initialization? Following the analysis of Jacot et al. (2018), the analogous modewise loss for a supervised setup takes the form $\mathcal{L}_j \propto (\bar{c}_j^* - \gamma_j c_j)^2$, with c_j a learnable coefficient of Gram matrix eigenvector j in the representer theorem coefficients of the learned function and $\bar{c}_j^*$ a constant. We nondimensionalize in this case as $\bar{c}_j = \gamma_j c_j$. As shown in Figure 8B, the landscape in the supervised case is merely quadratic and has no unstable equilibria, reflecting the lack of inversion symmetry in the loss. Therefore $\bar{c}_j'(0)$

[23] To be more precise, it takes a time $\tau_j^{(\epsilon)} \approx -\log(\epsilon/|\bar{s}_j(0)|)/4\gamma_j$ to escape a ball of small constant radius $\epsilon > 0$ around the origin, and we drop the sub-leading-order contribution of ϵ.

[24] Our model in fact obeys the more general symmetry $\boldsymbol{f} \to \boldsymbol{U}\boldsymbol{f}$ for any orthonormal matrix $\boldsymbol{U}$, which is shared by SimCLR but not by VICReg or Barlow Twins with $\lambda \neq 1$.

is not small, and the respective coefficients grow immediately rather than first undergoing a period of slow growth, as shown in Figure 8D.

The main surprising finding of our experiments is that SSL experiments with ResNets exhibit stepwise learning (especially upon small init) even far from the linear (i.e. lazy) regime. It thus seems likely that one could derive the stepwise phenomenon from a much looser set of assumptions on the model class. The view of stepwise learning as a simple consequence of SSB may be of use in the development of a more general theory in this sense. This SSB view suggests that we may understand the growth of each new embedding direction from zero as the escape from a saddle point and use the fact that the local loss landscape around any saddle point of a given index (and with nondegenerate Hessian) is the same as any other up to rescaling of the domain. We conjecture that stepwise learning will occur generically modulo edge cases (such as simultaneous growth of multiple directions) given an appropriate choice of loss function under minimal and realistic conditions on the empirical (i.e. time-varying) NTK of the model and assuming the final embeddings are the output of a linear readout layer.

G. Potential modifications for speeding up SSL

Compared to standard supervised learning, SSL is known in folklore to be much slower to train. Our work presents a theory of the training dynamics of SSL, and thus it is natural to ask whether it sheds light on this point of difficulty or suggests means by which it might be addressed. Here we suggest an explanation for the slowness of SSL training and propose various fixes which are ripe for future study.

In our picture of the training dynamics of SSL, embedding eigenvalues start small and grow sequentially, converging when d eigenvalues have sufficiently grown. Smaller eigendirections require more time to grow. **We suggest that SSL is slow to converge because one must wait for the small eigendirections to grow.** This hypothesis is supported by Figure 3, which shows that, in a realistic configuration, a significant fraction of the eigenvalues remain small even late in training.[25]

This suggests that SSL can be sped up by modifying training to target small eigenvalues. Here we suggest one way this might be achieved via preconditioning of gradients and two additional ways this might be achieved by modifying the loss function itself. We leave our results at the level of theoretical speculation. We encourage interested SSL practitioners to try implementing the methods described and reach out with questions.

G.1. Targeting gradients towards small PCA directions.

One potentially promising idea is to simply compute the PCA matrix of the embeddings and "manually" increase the gradient pull along directions which are very small, thereby encouraging the distribution to expand in directions in which it is currently flat. Let us denote by $\boldsymbol{F} \in \mathbb{R}^{2n\times d}$ the embeddings for a given data batch. Backpropagation will compute $\nabla_\theta \mathcal{L} = \text{Tr}[(\nabla_\theta \boldsymbol{F})^\top \nabla_{\boldsymbol{F}} \mathcal{L}]$. We may simply enact the substitution

$$\nabla_{\boldsymbol{F}} \mathcal{L} \longrightarrow (\nabla_{\boldsymbol{F}} \mathcal{L})(\boldsymbol{F}^\top \boldsymbol{F} + \alpha \boldsymbol{I}_d)^{-1} \tag{52}$$

with $\alpha > 0$ to suppress $\nabla_{\boldsymbol{F}} \mathcal{L}$ along directions of large variance and increase it along directions of small variance. We anticipate this will encourage faster growth of small eigenmodes. This trick could apply to any joint embedding method.

G.2. Sharpening at zero

This and the following method are specialized to the Barlow Twins loss. They will make use of visual depictions of the modewise loss landscape as described in Appendix F, and so we recommend reading that appendix first. They will both consider transformations $g(\boldsymbol{C})$ of the correlation matrix $\boldsymbol{C}$, where g acts pointwise on the eigenvalues of $\boldsymbol{C}$.

Recall that we found that the singular values of the system under aligned conditions evolve under a loss of the form $\mathcal{L}_j = (1-\lambda_j)^2 = (1-\gamma_j s_j^2)^2$. As shown in Figure 9A and discussed in Appendix F, this means that, when initialized near zero, they initially feel only a small gradient and must spend a long time escaping the origin, with that duration determined by just how small they are initially. However, this is no longer the case if we change the loss so it has a kink at zero. For example, if we change the loss to $\mathcal{L}_{\text{sharp}}(\boldsymbol{C}) = \|g_{\text{sqrt}}(\boldsymbol{C}) - \boldsymbol{I}_d\|_F$ with $g_{\text{sqrt}}(\lambda) = \text{sign}(\lambda)|\lambda|^{1/2}$, then all singular values with $\gamma_j > 0$ feel a $\Theta(1)$ repulsive force from the origin regardless of how close they are. This loss is plotted in Figure 9B.

[25] This observation is particularly compelling in light of the finding of Garrido et al. (2022) that generalization is better when embeddings have higher rank. Their work suggests that hyperparameters should be chosen to maximize the embedding rank at the end of training.

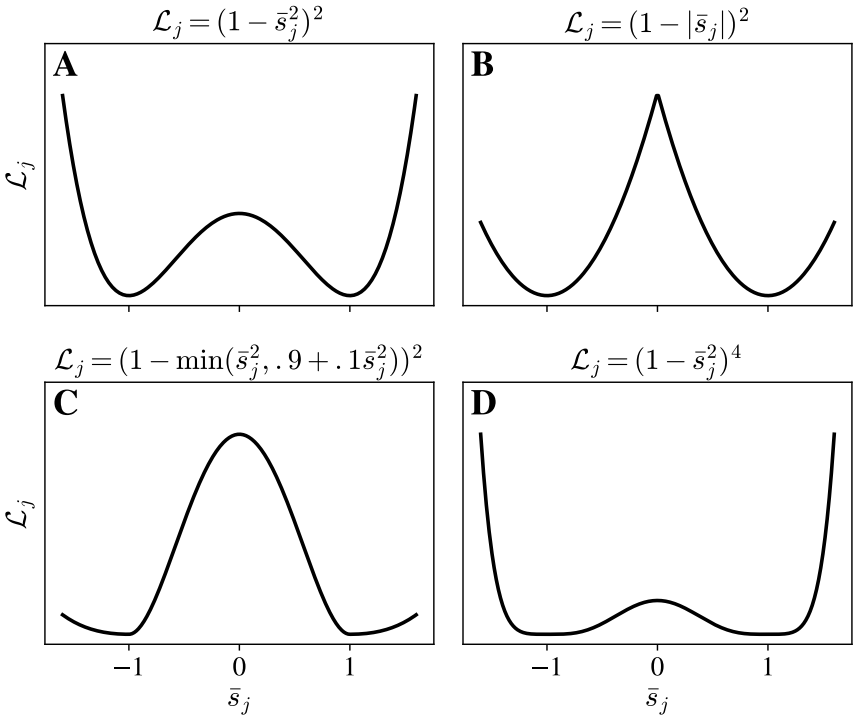

Figure 9: **Proposed modifications for the modewise SSL loss to encourage or enable the faster growth of slow eigendirections.** Losses are expressed as a function of $\bar{s}_j = \gamma_j^{1/2} s_j$ and assume $\gamma_j > 0$. **(A)**: Original Barlow Twins modewise loss. **(B)**: A modified loss with a kink at zero. The origin is no longer an unstable equilibrium and nearby points will quickly escape. **(C, D)**: Two losses modified so as to have smaller curvature at their minima and thereby permit larger learning rates without instability.

Interventions of this type lead to eigenvalue growth curves of the type shown in Figure 8D.[26]

G.3. Smoothing around minima

The naive way to speed up the growth of slow-growing eigenmodes is simply to increase the learning rate. This fails because the minima of the fast-growing eigenmodes will eventually become unstable. For example, if $d = 2$ and the total loss is $\mathcal{L} = (1 - s_1^2)^2 + (1 - 10^{-6}s_2^2)^2$, any learning rate above $\eta = 1/4$ will cause instability in s_1, but s_2 requires a larger learning rate to grow in reasonable time. One solution here is to modify the loss landscape so that the higher eigendirections see a wider basin around their minima and can thus tolerate a larger learning rate. One implementation might be $\mathcal{L}_{\text{smooth}}(\boldsymbol{C}) = \|g_{\text{smooth}}(\boldsymbol{C}) - \boldsymbol{I}_d\|_F$ with $g_{\text{smooth}}(\lambda) = \min(\lambda, 1)$. One might replace this with e.g. $g_{\text{smooth}} = \min(\lambda, 1 + \epsilon(1 - \lambda))$ for some small $\epsilon > 0$. Alternatively, one might modify the structure of the loss function itself to be e.g. $\mathcal{L}_j = (1 - \lambda_j)^4$, which has vanishing curvature at its minimum, or implement a similar idea with a hinge function to create a perfectly flat basin of finite width. Two possibilities for losses modified to have wider minima are shown in Figures 9C,D.

Both these last ideas are specialized to Barlow Twins in the case of $\lambda = 1$, but we anticipate they could be easily generalized. In fact, VICReg already includes a square root in the variance term which we conjecture implicitly does something similar to our proposed sharpening modification.

[26] In fact, the particular choice $g_{\text{sqrt}}(\lambda) = \text{sign}(\lambda)|\lambda|^{1/2}$ gives dynamics which can be solved exactly for aligned init just like the original Barlow Twins loss studied in the main text.