When Can We Approximate Wide Contrastive Models with Neural Tangent Kernels and Principal Component Analysis?

Gautham Govind Anil[1] *, **Pascal Esser**[2], **Debarghya Ghoshdastidar**[2]

[1]Indian Institute of Technology Madras
[2]Technical University of Munich
gauthamga.gga@gmail.com, {esser, ghoshdas}@cit.tum.de

Abstract

Contrastive learning is a paradigm for learning representations from unlabelled data and several recent works have claimed that such models effectively learn spectral embeddings and show relations between (wide) contrastive models and kernel principal component analysis (PCA). However, it is not known if trained contrastive models indeed correspond to kernel methods or PCA. In this work, we analyze the training dynamics of two-layer contrastive models, with non-linear activation, and answer when these models are close to PCA or kernel methods. It is well known in the supervised setting that neural networks are equivalent to neural tangent kernel (NTK) machines, and that the NTK of infinitely wide networks remains constant during training. We provide the first constancy results of NTK for contrastive losses, and present a nuanced picture: NTK of wide networks remains almost constant for cosine similarity based contrastive losses, but not for losses based on dot product similarity. We further study the training dynamics of contrastive models with orthogonality constraints on output layer, which is implicitly assumed in works relating contrastive learning to spectral embedding. Our deviation bounds suggest that representations learned by contrastive models are close to the principal components of a certain matrix computed from random features.

Introduction

The paradigm of self-supervised learning (SSL) builds on the idea of using knowledge about semantic similarities in the data to define which data-points should be mapped close to each other in the latent representation. The goal of SSL is to learn a "good representation". While there is no unique notion of "good" without taking a downstream task into consideration (Bengio, Courville, and Vincent 2013), in general one is interested in mapping semantically similar objects to close representations in the latent space, but avoid "dimension collapse" that occurs when different dimensions in the latent space *collapse* to the same value. Depending on the mechanism used to prevent collapse of learned embeddings, SSL strategies can be broadly categorised as contrastive or non-contrastive learning. Contrastive learning relies on negative samples to ensure representations do not collapse (Oord, Li, and Vinyals 2018; Chen et al. 2020; He et al. 2020; HaoChen et al. 2021), whereas non-contrastive learning avoids collapse by incorporating architectural asymmetry (Grill et al. 2020; Chen and He 2021) or reduction in dimension redundancy (Zbontar et al. 2021; Bardes, Ponce, and LeCun 2021). In practice, a plethora of SSL strategies, including deep contrastive and non-contrastive models, have been proposed over the past years across multiple domains; many of them demonstrating excellent performance empirically (Assran et al. 2022; Wang et al. 2023). While these works underline the importance of SSL and (non-)contrastive models for applications, their theoretical understanding is still limited.

Theoretical analysis of SSL is in its early stages. There has been considerable effort in deriving generalization error bounds for downstream tasks on learned embeddings (Arora et al. 2019b; Wei, Xie, and Ma 2021; Bao, Nagano, and Nozawa 2022), and analysing spectral / isoperimetric properties of data augmentation (Han, Ye, and Zhan 2023; Zhuo et al. 2023). Results based on learning theoretic measures (Saunshi et al. 2019; Wei et al. 2020; Nozawa and Sato 2021), information theory (Tsai et al. 2020; Tosh, Krishnamurthy, and Hsu 2021) and loss landscapes (Pokle et al. 2022; Ziyin et al. 2022) have been studied.

Generalisation bounds, however, provide little understanding of the representations learned via SSL. (Balestriero and LeCun 2022) answer this by showing that various (non-)contrastive learning formulations result in learning spectral embedding, principal component analysis (PCA) or their variants. In a similar vein, (Munkhoeva and Oseledets 2023) relate contrastive learning with trace maximization problems and matrix completion—all related to PCA. *Equivalences between the optimization formulations of SSL and PCA do not necessarily imply that (non-)contrastive models, trained with gradient descent, perform PCA*. This requires analysing either the converged solution or the training dynamics of SSL.

A number of works derive and study the training dynamics of (non)contrastive learning, albeit mostly limited to linear neural networks (Wang and Isola 2020; Tian, Chen, and Ganguli 2021; Wang and Liu 2021; Tian 2022; Esser, Mukherjee, and Ghoshdastidar 2023). In the context of non-linear networks, (Simon et al. 2023) suggest that for wide neural networks, that is, in the neural tangent kernel (NTK) regime (Jacot, Gabriel, and Hongler 2018; Lee

*Work done partly at Technical University of Munich

et al. 2019), contrastive learning could be equivalent to kernel PCA (Schölkopf, Smola, and Müller 1997). *Although no prior work explicitly analyzes the convergence of wide contrastive models to kernel (or NTK) machines, there has been a significant interest in training kernel models under (non)contrastive losses* (Kiani et al. 2022; Cabannes et al. 2023; Esser, Fleissner, and Ghoshdastidar 2023). Depending on the problem formulation, it can indeed be shown that these kernel contrastive models are closely related to kernel PCA (Esser, Fleissner, and Ghoshdastidar 2023) or kernel support vector machine (Shah et al. 2022).

Motivation and Contributions. In spite of strongly suggesting relations between constrative learning, PCA and kernel methods (or NTKs), existing theoretical works do not explicitly answer *if trained contrastive models are close to kernel methods, specifically with a fixed deterministic kernel* (as has been shown in the NTK regime for supervised models). There is also no theoretical evidence on *when trained contrastive models can be approximated by solutions of PCA or other trace maximization problems*. We analyse the training dynamics of two-layer non-linear networks trained under contrastive or non-contrastive losses, and rigorously answer both questions. Specifically:

1. We firstly derive the NTK of two-layer networks of width M trained under (non)-contrastive loss, and study the deviation between NTK after several steps of gradient descent from the NTK at initialization. Our results address questions on the constancy of NTK.

   Observation 1: (Non-)Contrastive losses are defined in terms of similarities between learned representations. We show that if the losses are in terms of dot-product similarity, then NTK drastically changes within $O(\log M)$ training time. Experiments on non-contrastive learning suggest that NTK changes (Simon et al. 2023), but there was no prior theoretical evidence.

   Observation 2: In contrast to dot product similarity, if the losses are defined in terms of cosine similarity—considered in *InfoNCE* (Oord, Li, and Vinyals 2018) and *SimCLR* (Chen et al. 2020)—then NTK after $O(M^{1/6})$ steps is close to NTK at initialization. Thus, contrastive models trained under such losses can be approximated by kernel methods, with a fixed NTK. Unfortunately, unlike supervised learning—where trained neural networks in NTK regime is the solution of kernel regression—there may not be a closed formed analytical solution of the trained model.

2. Secondly we study the training dynamics of (Grassmannian) gradient descent under orthogonality constraints of the output layer of the network. While orthogonality is not imposed in practical SSL approaches, it is often assumed in theoretical works to relate contrastive learning to variants of PCA (Munkhoeva and Oseledets 2023), in kernel SSL formulations (Esser, Fleissner, and Ghoshdastidar 2023), to prevent dimension collapse (Esser, Mukherjee, and Ghoshdastidar 2023) etc.

   Observation 1: We note that, with orthogonality constraint, some contrastive losses (or their modifications) are equivalent to PCA of a $M \times M$ matrix $\boldsymbol{C}(t)$ that depends on the non-linear features at the hidden layer, learned after t iterations of gradient descent.

   Observation 2: For some cosine-similarity based contrastive losses, the Frobenius norm deviation $\|\boldsymbol{C}(t) - \boldsymbol{C}(0)\|_F = O(t/\sqrt{M})$ suggesting that, in this case, wide contrastive models are close to PCA of a randomly initialised matrix $\boldsymbol{C}(0)$. Furthermore, the representation learned via PCA from $\boldsymbol{C}(t)$ and $\boldsymbol{C}(0)$ are also close, upto orthonormal rotations.

All proofs as well as additional empirical validations are provided in the supplementary material together with the code to reproduce the experiments.

Preliminaries and Problem Setup

Before going into the main results of the paper, we first outline the contrastive learning setup, the embedding function and NTK formulation under consideration, together with the general conditions for the NTK to remain constant during training. We use the following notation throughout the paper:

Notation. We use lowercase bold letters (e.g. $\boldsymbol{a}$) to denote vectors and upper case bold letters (e.g. $\boldsymbol{A}$) to denote matrices. Let $\boldsymbol{A}_{i\cdot}$ denote the i^{th} row and $\boldsymbol{A}_{\cdot i}$ denote the i^{th} column of matrix $\boldsymbol{A}$. Let $\mathbb{I}$ be an appropriately sized identity matrix. $\|\cdot\|_p$ denotes the L_p norm, $\|\cdot\|_F$ denotes the Frobenius norm and $\|\boldsymbol{A}\|_{\max} := \max_{ij}\{|\boldsymbol{A}_{ij}|\}$. We denote parameter $\boldsymbol{\Theta}$ at time-step t by $\boldsymbol{\Theta}(t)$; however the time indexing is suppressed when it is clear from the context to improve readability.

(Non-)Contrastive Learning

In this work, our primary focus is on sample-contrastive methods which use multiple positive/ negative sample pairs. Consider a dataset of N datapoints: $\mathcal{D} := \left\{\left\{\boldsymbol{x}_n, \boldsymbol{x}_{n,q}\right\}_{q=1}^{Q}\right\}_{n=1}^{N}$, where $\boldsymbol{x}_n \in \mathbb{R}^D$ denotes the n^{th} D dimensional data sample and $\boldsymbol{x}_{n,q}$ denotes the q^{th} pair in relation to $\boldsymbol{x}_n$.[1] Using this formulation, we now state a general form for the contrastive loss:

$$\mathcal{L}(\mathcal{D}) := \frac{1}{N}\sum_{n=1}^{N} l\left(\left\{s\left(\boldsymbol{x}_n, \boldsymbol{x}_{n,q}\right)\right\}_{q=1}^{Q}\right) \tag{1}$$

where $l(\cdot)$ is some function and $s(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is the similarity between representations of inputs $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ learned by a (non-)contrastive model. While *softmax* or its logarithm are typically used for $l(\cdot)$ in practice, theoretical works often consider $l(\cdot)$ to be *linear* (Ji et al. 2023; Esser, Mukherjee, and Ghoshdastidar 2023). While a wide range of similarity measures $s(\cdot,\cdot)$ are considered, they often build on similar underlying ideas. Losses such as *MoCo* (He et al. 2020) build

[1] Note that the pair could involve a positive or negative sample. Hence, this framework encompasses popular examples such as the *contrastive triplet* setting $\{\boldsymbol{x}_n, \boldsymbol{x}_n^+, \boldsymbol{x}_n^-\}_{n=1}^N$ and the *non-contrastive* setting $\{\boldsymbol{x}_n, \boldsymbol{x}_n^+\}_{n=1}^N$.

on *dot product similarity*, while the popular *SimCLR* and *InfoNCE* (Chen et al. 2020; Oord, Li, and Vinyals 2018) losses build on *cosine similarity*. Therefore, we consider the following two similarity measures, where $\boldsymbol{x} \mapsto f(\boldsymbol{x})$ denotes the learned representation:[2]

$$s(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = f(\boldsymbol{x})^\top f(\tilde{\boldsymbol{x}}), \qquad \text{(dot product)}$$

$$s(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \frac{f(\boldsymbol{x})^\top f(\tilde{\boldsymbol{x}})}{(\|f(\boldsymbol{x})\| + \delta)(\|f(\tilde{\boldsymbol{x}})\| + \delta)}. \qquad \text{(cosine)}$$

We consider the following set of assumptions on the similarity measure and on the data:

Assumption 1 (Constant for cosine similarity)**.** *δ is a small strictly positive constant.*

Assumption 2 (Smoothness)**.** $\left|\frac{\partial l(\,\cdot\,)}{\partial s(\boldsymbol{x}, \tilde{\boldsymbol{x}})}\right| \leq c_l \;\forall\; \boldsymbol{x}, \tilde{\boldsymbol{x}}$.

Assumption 3 (Bounded inputs)**.** *Inputs are bounded,* $\max_{n,q} \left\{ \|\boldsymbol{x}_n\|_\infty, \|\boldsymbol{x}_{n,q}\|_\infty \right\} \leq c_{in}$.

While δ is not typically considered in cosine similarity, assuming $\delta > 0$ ensures that $s(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is defined even when norms of the representations are zero. Furthermore, $\delta > 0$ can be made arbitrarily small, making Assumption 1 practically reasonable. Apart from making the cosine similarity computation numerically stable, this structure for cosine similarity helps to simplify the proofs by providing a strictly positive lower bound on $\|f(\boldsymbol{x})\| + \delta$ for any $\boldsymbol{x}$. Assumption 2 is evidently satisfied for commonly considered losses where $l(\cdot)$ is *linear* or *softmax* . Assumption 3 is often considered for theoretical analysis in NTK literature (e.g. (Jacot, Gabriel, and Hongler 2018)).

Embedding Function

The outlined setup for contrastive losses is stated for an arbitrary embedding function $f(\cdot)$. However, for our analysis, we focus on one hidden layer neural networks. We aim to find a mapping $f(\boldsymbol{x}; \theta) : \mathbb{R}^D \to \mathbb{R}^Z$ parameterized by θ where typically $D > Z$. In particular, we consider a two-layer fully connected non-linear neural network: $f(\boldsymbol{x}; \theta) = \boldsymbol{W}^\top \phi(\boldsymbol{V}\boldsymbol{x})$ where $\boldsymbol{x} \in \mathbb{R}^D$ is an input vector and ϕ is a pointwise non-linear activation function. $\boldsymbol{W} \in \mathbb{R}^{M \times D}$ and $\boldsymbol{V} \in \mathbb{R}^{M \times Z}$ are the trainable weight matrices. Let θ be the vector which contains all entries of $\boldsymbol{W}$ and $\boldsymbol{V}$. In the context of (infinite) width analysis, the 'appropriate' initialization of these weights is essential. Existing NTK literature on supervised learning (e.g. (Jacot, Gabriel, and Hongler 2018; Arora et al. 2019a)) considers the following parameterization:

$$f(\boldsymbol{x}; \theta) := M^{-1/2} \boldsymbol{W}^\top \phi(\boldsymbol{V}\boldsymbol{x}) \tag{2}$$

where each $\boldsymbol{W}_{i,j}, \boldsymbol{V}_{i,j} \sim \mathcal{N}(0, 1)$. We consider this setup, termed *NTK parametrization*, for the remainder of the paper. We additionally assume:

Assumption 4 (Smoothness of activation function)**.** *ϕ is L_ϕ-Lipschitz and β_ϕ-smooth.*

[2]Note that $f(\cdot)$ is a parameterized function as we later define in (2). However, we suppress the parameterization here for ease of notation.

Assumption 4 is a usually considered smoothness criterion (and holds for sigmoid, tanh etc.). We additionally note:

Lemma 5 (Weights at initialization are bounded)**.** *For any $\delta \in (0, 1)$, there exists width-independent constants $c_\theta, c_s > 0$ s.t. w.p. $1 - \delta$: $\|\boldsymbol{W}(0)\|_{\max}, \|\boldsymbol{V}(0)\|_{\max} \leq c_\theta \log M$; $\|\boldsymbol{W}(0)\|_2, \|\boldsymbol{V}(0)\|_2 \leq c_s\sqrt{M}$.*

Lemma 6 (Bounds on gradients and weights)**.** *Let $\boldsymbol{b}_1 = \frac{\partial f}{\partial \boldsymbol{x}}$, $\boldsymbol{b}_2 = \frac{\partial f}{\partial \phi(\boldsymbol{V}\boldsymbol{x})}$. For any $\delta \in (0, 1)$, there exists width-independent constants $c_s, s_0 > 0$ s.t. at initialization w.p. $1 - \delta$: $\|\boldsymbol{W}(0)\|_2$, $\|\boldsymbol{V}(0)\|_2 \leq c_s\sqrt{M}$ and $\|\boldsymbol{b}_i\|_\infty \leq \frac{s_0}{\sqrt{M}} \|\boldsymbol{b}_i\|_2$, $i = 1, 2$.*

Lemmas 5 and 6 hold w.h.p. for wide networks under the considered Gaussian initialization (see Liu, Zhu, and Belkin 2020b,a).

Conditions for Constancy of NTK

Let us start by outlining the NTK analysis in general for a function $f(\boldsymbol{x}; \theta(t)) : \mathbb{R}^D \to \mathbb{R}^Z$, where $Z \geq 1$. For input vectors $\boldsymbol{x} \in \mathbb{R}^D$ and $\tilde{\boldsymbol{x}} \in \mathbb{R}^D$, we define the *empirical NTK* for a neural network $f(\cdot)$ parameterized by $\theta(t)$ as $\boldsymbol{K}_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \theta(t)) := \frac{\partial f_i(\boldsymbol{x}; \theta(t))}{\partial \theta(t)}^\top \frac{\partial f_j(\tilde{\boldsymbol{x}}; \theta(t))}{\partial \theta(t)}$ where $f_i(\boldsymbol{x}; \theta)$ represent the i^{th} entry of the Z dimensional function output. In general, $\boldsymbol{K}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \theta(t))$ varies with time t as the model is trained. However, under certain conditions, for infinitely wide neural networks, the NTK stays constant during training (Jacot, Gabriel, and Hongler 2018; Arora et al. 2019a), i.e as $M \to \infty$:

$$\forall t \quad |\boldsymbol{K}_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \theta(t)) - \boldsymbol{K}_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \theta(0))| \to 0. \tag{3}$$

Furthermore, under Gaussian initialization of parameters, it holds that the NTK at initialization converges, as $M \to \infty$, to an *analytical NTK* $\boldsymbol{K}^*_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) := \mathbb{E}_\theta \left[\boldsymbol{K}_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \theta)\right]$. As the NTK does not change during training, the training dynamics of the network at any time step can be written in terms of $\boldsymbol{K}^*$ — in the supervised setting, this leads to kernel regression at convergence. To prove constancy of the form (3), several works have analyzed NTKs in the supervised setting (Arora et al. 2019a; Lee et al. 2019; Chizat, Oyallon, and Bach 2019). In particular, (Liu, Zhu, and Belkin 2020a) showed that the constancy of NTK is predicated on the spectral norm of the Hessian.

To study the NTK of contrastive models, we consider $f(\boldsymbol{x}; \theta)$ of the form (2) and use the machinery built in (Liu, Zhu, and Belkin 2020a,b). Define Z Hessian matrices, one for each element of the output representation. The z^{th} Hessian matrix $\boldsymbol{H}^{(z)}$ (evaluated at input $\boldsymbol{x}$) is $\boldsymbol{H}^{(z)}_{ij}(\boldsymbol{x}) := \frac{\partial^2 f_z(\boldsymbol{x}; \theta(t))}{\partial \theta_i(t) \partial \theta_j(t)}$, $z \in [Z]$. We bound the change in the spectral norm of the Hessian in terms of the change in weights by adapting Theorem 7.1 of (Liu, Zhu, and Belkin 2020b)[3] to account for multi-dimensional outputs:

[3]Theorem 7.1 of (Liu, Zhu, and Belkin 2020b) gives a bound of the form $\left\|\boldsymbol{H}^{(z)}(\boldsymbol{x}; \theta(t))\right\|_2 = O\left(\frac{R^{3L}}{\sqrt{M}}\right)$ for a network with L layers. However, for two-layer networks, it is possible to reduce this bound to the form given in Lemma 7.

Lemma 7. (Bound on the norm of the Hessian) *Under Assumptions 3, 4, consider the neural network defined in* (2)*. The following holds w.p.* $1-\delta, \delta \in (0,1)$*: If the change in weights during training is bounded as*

$$\|\boldsymbol{W}(t) - \boldsymbol{W}(0)\|_F + \|\boldsymbol{V}(t) - \boldsymbol{V}(0)\|_F \leq R, \quad (4)$$

then, $\forall\ z \in [Z]$*, with* $\alpha_1 = 4\beta_\phi c_{in}^2 L_\phi$ *and* $\alpha_2 = 4L_\phi c_{in}(1 + \beta_\phi c_{in} s_0 c_s)$*, the* z^{th} *Hessian is bounded as:* $\left\|\boldsymbol{H}^{(z)}(\boldsymbol{x}; \theta(t))\right\|_2 \leq \frac{\alpha_1 R + \alpha_2}{\sqrt{M}}$.

With help of Lemma 7, we can now bound the change in NTK. Towards this, we extend Proposition 2.3 of (Liu, Zhu, and Belkin 2020a) to the multi-dimensional case ($Z > 1$) to obtain the following lemma:

Lemma 8. (Bound on the change in NTK) *Define* $\mathbb{S} := \{\boldsymbol{s} \in \mathbb{R}^p; \|\boldsymbol{s} - \boldsymbol{s}(0)\| \leq R\}$*, where* p *is the total number of learnable parameters in* (2)*. Assume that for any input* $\boldsymbol{x}$*,* $\left\|\boldsymbol{H}^{(z)}(\boldsymbol{x}; \boldsymbol{s})\right\|_2 \leq \epsilon$ *and* $\|\nabla_{\boldsymbol{s}} f_z(\boldsymbol{x}; \boldsymbol{s})\|_2 \leq c_0$*,* $\forall\ z \in [Z]$ *and* $\forall\ \boldsymbol{s} \in \mathbb{S}$*. Then, for any inputs* $\boldsymbol{x}, \tilde{\boldsymbol{x}}$*,* $\forall\ \boldsymbol{s} \in \mathbb{S}$ *and* $\forall\ i, j \in [Z]$*,* $|\boldsymbol{K}_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \boldsymbol{s}) - \boldsymbol{K}_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \boldsymbol{s}(0))| \leq 2\epsilon c_0 R$*.*

If $\boldsymbol{K}$ does not change during training, the analytical (expected) NTK $\boldsymbol{K}^*$ models the behaviour of the network not only at initialization, but also at convergence and therefore allows us to express the network dynamics in a simple form. In supervised settings with squared loss, it is known that condition (4) is true till convergence for wide neural networks (Liu, Zhu, and Belkin 2020b). While it is possible to use Lemmas 7 and 8 to examine the behaviour of NTK in general, note that it is **not known when the condition** (4) **holds for an arbitrary loss function**. In the following Section, we study the validity of this condition when learning embeddings using aforementioned (non-)contrastive losses.

Constancy of NTKs Under Contrastive Losses

We now examine the NTK evolution for neural networks trained under contrastive losses. To do so, using the above presented setup, we derive the dynamics of a neural network trained using gradient flow under a loss of the form (1) in terms of the NTK of the neural network as defined in (2):

Lemma 9. (Contrastive dynamics in terms of NTK) *Consider training a neural network of the form (2) using a loss* $l(\cdot)$ *of the form (1) under gradient flow on dataset* $\mathcal{D}$*. Let* $g_i^{(t)}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) := \frac{\partial s(\boldsymbol{x}, \tilde{\boldsymbol{x}})}{\partial f_i(\boldsymbol{x}; \theta(t))}$ *and* $\boldsymbol{K}_{\cdot}^{(t)}(\cdot, \cdot) := \boldsymbol{K}_{\cdot}(\cdot, \cdot; \theta(t))$*. Then, for* $z \in [Z]$*, the representation of an arbitrary input* $\tilde{\boldsymbol{x}}$ *evolves as:* $-\frac{1}{N}\sum_{n,q} \frac{\partial l(\cdot)}{\partial s(\boldsymbol{x}_n, \boldsymbol{x}_{n,q})} \left[\sum_{i=1}^{Z} \left[\boldsymbol{K}_{zi}^{(t)}(\tilde{\boldsymbol{x}}, \boldsymbol{x}_n) g_i^{(t)}(\boldsymbol{x}_n, \boldsymbol{x}_{n,q}) + \boldsymbol{K}_{zi}^{(t)}(\tilde{\boldsymbol{x}}, \boldsymbol{x}_{n,q}) g_i^{(t)}(\boldsymbol{x}_{n,q}, \boldsymbol{x}_n)\right]\right]$.

While Lemma 9 holds for any loss of the form (1), we are interested in the behaviour of the NTK when trained under losses which use dot product or cosine similarity as the similarity measure. Empirical validation of Lemma 9 is provided in the supplemental material.

NTK for Dot product Does Not Stay Constant

We start with examining the change in weights (as defined in (4)) of a network trained using contrastive losses which utilize dot product as the similarity measure. In particular, we show that in this setting, there exists cases where the change in weights become arbitrarily large even for arbitrarily wide neural networks, implying that the NTK does not remain constant. To demonstrate this, we consider a simple loss function of the form (1) under dot product similarity. Because there is no normalization in the dot product similarity measure, the loss can be minimized arbitrarily by scaling the weights and hence we expect the weights to grow arbitrarily large with time. We formalize this notion and its implications on the constancy of the NTK in the following proposition:

Proposition 10. (NTK under dot product does not remain constant) *For* $D = Z = 1$*, linear loss* $(l(a) := a)$*, dot product similarity and triplet setting* $(\mathcal{D} = \{x_n, x_n^+, x_n^-\}_{n=1}^N)$ *in* (1)*, the optimization is:* $\min_\theta \frac{1}{N}\sum_{n=1}^N f(x_n; \theta)\left(f(x_n^-; \theta) - f(x_n^+; \theta)\right)$*. Consider a network* (2) *with linear activation* $(\phi(a) := a)$*, weights initialised as independent* $\mathcal{N}(0,1)$*, and trained via gradient flow.*
There is a dataset such that, with probability at least $1 - \frac{25}{\sqrt{M}}$*, for a time step* $\tilde{t} \in (0, \log M)$ *and any input pair* $x, \tilde{x}$ *with* $x\tilde{x} \neq 0$*, the NTK satisfies* $|\boldsymbol{K}(x, \tilde{x}; \theta(t)) - \boldsymbol{K}(x, \tilde{x}; \theta(0))| \to \infty$ *as* $t \to \tilde{t}$*.*

Proposition 10 shows there are cases where the *NTK does not remain constant* even for arbitrarily wide networks and logarithmic training time when trained under dot product similarity based loss. Empirical validation of Proposition 10 is provided in the supplemental material.

NTK for Cosine similarity Losses Stays Constant

Considering the same question as in the previous section, we now shift our focus onto the constancy of NTK for losses defined in terms of the cosine similarity. The key difference between dot product and cosine similarity is the presence of *normalization by norms of the representations*. We now show that this normalization plays an important role in deciding the learning dynamics and examine its implications on the constancy of the NTK. To prove constancy of the NTK, we make use of the fact that the similarity measure is normalized and first establish a bound on the maximum element-wise change in weights.

Lemma 11. (Bound on element-wise change in weights under cosine similarity) *The following holds w.p.* $1-\delta, \delta \in (0,1)$*. Under Assumptions 1 - 4, consider losses of the form* (1) *where cosine similarity is used. If a neural network* $f(\cdot)$ *as defined in* (2) *is trained using gradient descent with learning rate* η*, at any time* t*, the change in weights are bounded as:* $|\Delta \boldsymbol{V}_{ij}(t)| \leq \frac{\beta_1}{\sqrt{M}} \|\boldsymbol{W}(t)\|_{\max}$ *and* $|\Delta \boldsymbol{W}_{ij}(t)| \leq \frac{\beta_2}{\sqrt{M}} \|\boldsymbol{V}(t)\|_{\max}$ *where* $\beta_1 = \frac{4\eta}{\delta} c_l c_{in} Q \sqrt{Z} L_\phi$ *and* $\beta_2 = \frac{4\eta}{\delta} c_l c_{in} Q D L_\phi$ *are constants independent of* M*.*

From Lemma 11, it can be seen that bounds for change in $\boldsymbol{V}$ and $\boldsymbol{W}$ form a coupled system. To study the discrete-time

dynamics of this system, we define and characterize a useful quantity $c(t)$:

Lemma 12. (Bound on weight difference while training under cosine similarity) *Let* $\beta := \max\{\beta_1, \beta_2\}$ *and* $c(t) := c(0)\left(1 + \frac{\beta}{\sqrt{M}}\right)^t$ *where* $c(0) = c_\theta \log M$. *Then, for any t, we have :* $\|\boldsymbol{V}(t) - \boldsymbol{V}(0)\|_{\max} \leq c(t) - c(0)$ *and* $\|\boldsymbol{W}(t) - \boldsymbol{W}(0)\|_{\max} \leq c(t) - c(0)$.

We now state the main theorem regarding the convergence of NTK for cosine similarity losses:

Theorem 13. (Bound on the change in NTK under cosine similarity) *The following holds w.p.* $1 - \delta, \delta \in (0, 1)$. *Consider losses of the form* (1) *with cosine similarity. Let* $c(0), \beta$ *be the constant in Lemma 12,* R *be as in* (4)[4]*,* α_1, α_2 *be as in Lemma 7 and* $\gamma := 2\sqrt{2}DL_\phi c_{in}$. *If a neural network* $f(\cdot)$ *of the form* (2) *is trained using gradient descent, then under Assumptions 1 - 2, for* $t \leq M^\alpha$ *iterations, the change in NTK is bounded as* $|\boldsymbol{K}_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \theta(t)) - \boldsymbol{K}_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \theta(0))| \leq \gamma\left(c(0)\exp(\beta M^{\alpha-0.5})\right)\frac{1}{\sqrt{M}}(\alpha_1 R^2 + \alpha_2 R)$. *In particular, if we set* $\alpha = \frac{1}{6}$ *and assume* $M \geq \max\{1, \beta^3\}$, *then the above statement simplifies to* $\max_{t\in(0,M^{1/6}]} \sup_{\boldsymbol{x},\tilde{\boldsymbol{x}}} |\boldsymbol{K}_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \theta(t)) - \boldsymbol{K}_{ij}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; \theta(0))| = O\left(M^{-1/6}(\log M)^3\right)$.

According to Theorem 13, wide neural networks trained under cosine similarity based contrastive loss have a nearly constant NTK even after $M^{1/6}$ iterations of gradient descent. This is in sharp contrast to networks trained under dot product based losses where the change in weights become arbitrarily large within $\log M$ gradient descent updates. Intuitively, this holds since normalization ensures that the change in weights remain sufficiently small in the case of cosine similarity. Empirical validation of Theorem 13 is provided in the supplemental material.

Remark 14. (Closed form solution in terms of NTK) *In the supervised setting, combining the analytical NTK with the closed form solution for kernel regression provides a closed form expression of the network trained until convergence (Jacot, Gabriel, and Hongler 2018), or even when early stopped. We take a step towards such a result for contrastive losses. In Lemma 9, we present the learning dynamics in the contrastive loss setting in terms of the NTK and in Theorem 13, we show that the NTK remains constant during training for* $M^{1/6}$ *steps under cosine similarity. While this is an important step towards a better theoretical understanding of contrastive models, it does not yet provide a closed form solution of the model output in terms of the NTK. While prior works on kernel contrastive methods suggest that, in the wide neural network regime, contrastive losses could be equivalent to kernel PCA (Simon et al. 2023), this connection has not been proven so far and is not apparent from the dynamics derived in Lemma 9 even if the NTK is constant (for cosine similarity based losses). Therefore, we shift our viewpoint from NTK dynamics to explicitly investigating this connection.*

[4]Note that R here is a function of M, with the relation being given by Lemma 12. Similarly, $c(0) = c_\theta \log M$.

From Wide Contrastive Models to PCA

We next study if there is a formal connection between PCA and representations learned by contrastive models, trained with gradient descent. Prior works have connected contrastive models to a trace maximization problem with an orthogonality constraint on the output layer, of the form:

$$\max_{\boldsymbol{W},\vartheta} \operatorname{Tr}\left(\boldsymbol{W}^\top \boldsymbol{C}_\vartheta \boldsymbol{W}\right) \quad \text{s.t.} \quad \boldsymbol{W}^\top \boldsymbol{W} = \mathbb{I}_Z, \tag{5}$$

where $\boldsymbol{C}_\vartheta \in \mathbb{R}^{M\times M}$ is a symmetric matrix that has a (possibly non-linear) data dependence through a function parameterized by ϑ. If $\boldsymbol{C}_\vartheta$ stays constant during optimization, optimization is done only over $\boldsymbol{W}$ and hence the problem simplifies to PCA on $\boldsymbol{C}_\vartheta$. To connect contrastive losses and PCA, it is then necessary to analyze the behaviour of $\boldsymbol{C}_\vartheta$ when a contrastive model is trained. Existing works do not examine this aspect of neural network dynamics. More specifically, (Esser, Fleissner, and Ghoshdastidar 2023) considers a kernel setting where learning is done using a contrastive loss of the form (5), but does not link it to the neural network dynamics. (Esser, Mukherjee, and Ghoshdastidar 2023) considers neural network dynamics for contrastive losses of the form (5) but only examines the linear setting. (Simon et al. 2023) considers the dynamics for kernel models but does not investigate if the kernel dynamics are close to the neural network dynamics. (Munkhoeva and Oseledets 2023) reformulates losses such as *SimCLR* to (5).

Extending prior works, we work towards formalizing the connection between **non-linear, wide** networks and PCA, not only through rewriting the loss, but also by taking **learning into consideration**. Let us consider (1) with a linear loss such that we obtain (6). In addition, we consider (2) under orthogonality constraint on the second layer to obtain a neural network of the form (7):

$$\mathcal{L}(\mathcal{D}) := -\frac{1}{N}\sum_{n=1}^{N}\sum_{q=1}^{Q}\alpha_q s(\boldsymbol{x}_n, \boldsymbol{x}_{nq}) \tag{6}$$

$$f^\perp(\boldsymbol{x}; \theta) := M^{-1/2}\boldsymbol{W}^\top\phi(\boldsymbol{V}\boldsymbol{x}) \quad \text{s.t.} \boldsymbol{W}^\top\boldsymbol{W} = \mathbb{I}_Z. \tag{7}$$

where $\alpha_q = 1$ if $(\boldsymbol{x}_n, \boldsymbol{x}_{nq})$ is a positive pair and $\alpha_q = -1$ for a negative pair. Observe that if we use the dot product similarity, $s(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = f^\perp(\boldsymbol{x}; \theta)^\top f^\perp(\tilde{\boldsymbol{x}}; \theta)$, then the minimisation of the contrastive loss $\mathcal{L}(\mathcal{D})$ in (6) can be directly posed as a trace maximization problem[5] (5) with $\boldsymbol{C}_\vartheta = \widetilde{\boldsymbol{C}}_V = \frac{\boldsymbol{C}_V + \boldsymbol{C}_V^\top}{2}$;

$$\boldsymbol{C}_V = \frac{1}{MN}\sum_{n=1}^{N}\sum_{q=1}^{Q}\alpha_q\phi(\boldsymbol{V}\boldsymbol{x}_{n,q})\phi(\boldsymbol{V}\boldsymbol{x}_n)^\top. \tag{8}$$

Remark 15. (From contrastive models to PCA in case of dot product similarity) *Proposition 10 shows that the NTK diverges within* $\log(M)$ *steps for training with dot product based losses. While we do not explicitly prove this, it also*

[5]The value of the Trace in (5) is the same irrespective of whether we use $\boldsymbol{C}_V$ or $\widetilde{\boldsymbol{C}}_V$. However, using the symmetric $\widetilde{\boldsymbol{C}}_V$ allows us to make the connection of solving (5) through PCA more direct.

follows that, as the first layer $\boldsymbol{V}$ is trained, the matrix $\boldsymbol{C}_{\boldsymbol{V}}(t)$ diverges arbitrarily from the initialization $\boldsymbol{C}_{\boldsymbol{V}}(0)$. Hence, while minimizing $\mathcal{L}(\mathcal{D})$ for any fixed $\boldsymbol{V}$ corresponds to PCA for finding $\boldsymbol{W}$ (see (8)*), contrastive models trained with dot product based losses do not seem to be equivalent to PCA for a constant matrix $\boldsymbol{C}_{\vartheta}$.*

Cosine similarity Objective is Close to PCA for Wide Networks

Due to the near constancy of NTK under cosine similarity based losses, we investigate if it is possible to relate cosine similarity based trained contrastive models with PCA of a fixed matrix, potentially $\boldsymbol{C}_{\vartheta} = \widetilde{\boldsymbol{C}}_{\boldsymbol{V}}(0)$. A direct equivalence seems complicated due to the normalization terms of cosine similarity. However, we note that with orthogonality on $\boldsymbol{W}$, the cosine similarity measure can be bounded from below as

$$\frac{\left(\boldsymbol{W}^\top \phi(\boldsymbol{V}\boldsymbol{x})\right)^\top \left(\boldsymbol{W}^\top \phi(\boldsymbol{V}\tilde{\boldsymbol{x}})\right)}{(\|\phi(\boldsymbol{V}\boldsymbol{x})\| + \delta')(\|\phi(\boldsymbol{V}\tilde{\boldsymbol{x}})\| + \delta')} \tag{9}$$

since $\left\|\boldsymbol{W}^\top \phi(\boldsymbol{V}\boldsymbol{x})\right\|_2 \leq \left\|\boldsymbol{W}^\top\right\|_2 \|\phi(\boldsymbol{V}\boldsymbol{x})\|_2$ and $\|\boldsymbol{W}\|_2 = 1$ (where $\delta' := \sqrt{M}\delta$). This fact can now be used to define a modified loss $\mathcal{L}(\mathcal{D})$, where $s(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is defined as the bound given by (9). Minimizing the corresponding loss (6) with (7) can now be written as (5) with $\boldsymbol{C}_{\vartheta} = \widetilde{\boldsymbol{C}}_V = \frac{\boldsymbol{C}_V + \boldsymbol{C}_V^\top}{2}$ and define $\boldsymbol{C}_V$ as

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{q=1}^{Q} \frac{\alpha_q \phi(\boldsymbol{V}\boldsymbol{x}_{n,q})\phi(\boldsymbol{V}\boldsymbol{x}_n)^\top}{(\|\phi(\boldsymbol{V}\boldsymbol{x}_{n,q})\| + \delta')(\|\phi(\boldsymbol{V}\boldsymbol{x}_n)\| + \delta')} \tag{10}$$

Note that (10) is of the form (5), however, $\boldsymbol{C}_V$ is still dependent on $\boldsymbol{V}$ and hence the optimization in (6) with (7) is performed over both $\boldsymbol{V}$ and $\boldsymbol{W}$. Lemma 16 below shows that for wide networks trained with cosine similarity based losses, $\boldsymbol{C}_V(0)$ is close to $\boldsymbol{C}_V(t)$ in Frobenius norm. Hence $\widetilde{\boldsymbol{C}}_V$ also remains almost constant, which intuitively suggests that (10) becomes "close" to (5).

Lemma 16 (Constancy of $\boldsymbol{C}_V(t)$)**.** *Under Assumptions 1–4 and constraint $\boldsymbol{W}^\top \boldsymbol{W} = \mathbb{I}_Z$, consider training $f^{\perp}(\cdot)$ in* (7) *for t iterations using Grassmannian gradient descent*[6] *under losses of the form* (10) *with learning rate η. Then $\|\boldsymbol{C}_V(t) - \boldsymbol{C}_V(0)\|_F \leq \kappa \frac{t}{\sqrt{M}}$, where $\kappa := 16\delta^{-2}\eta Q^2 L_\phi^2 c_{in}^2 \sqrt{D}$.*

Numerical Simulation on MNIST. We train a network of the form (7) using a loss of the form (10). We then examine the evolution of the quantity $\|\boldsymbol{C}_V(t) - \boldsymbol{C}_V(0)\|_F$ with training across varying widths. The results are shown in Figure 1 (a), where colors indicate the epochs ($t \in [500]$). While the difference increases slightly with training, it goes down roughly as $\frac{1}{\sqrt{M}}$ with an increase in width, which is in line with the behaviour predicted in Lemma 16. In addition, we observe in Figure 1 (b) that $\boldsymbol{W}(t)$ changes significantly faster than $\boldsymbol{C}_V(t)$ during training; this suggests that the $\boldsymbol{W}$ that is learned is indeed the PCA of a $\widetilde{\boldsymbol{C}}_V(t)$ that is close to $\widetilde{\boldsymbol{C}}_V(0)$.

[6] In short, following (Edelman, Arias, and Smith 1998), the derivative of a function $g(\cdot)$ restricted to a Grassmannian manifold can be obtained by left-multiplying $1 - g(\cdot)g(\cdot)^\top$ to the unrestricted derivative of $g(\cdot)$.

Representations learned from PCA of $\widetilde{\boldsymbol{C}}(0)$ and PCA of $\widetilde{\boldsymbol{C}}(t)$ are close

We characterize the difference between the representations learned by performing PCA on $\widetilde{\boldsymbol{C}}_V(0)$ and on $\widetilde{\boldsymbol{C}}_V(t)$. Lemmas 16–17 suggest that training (7) under (6) could be close to PCA on $\boldsymbol{C}_V(0)$.

Lemma 17. (Perturbation bound on representation) *Let $u(\boldsymbol{x}; \boldsymbol{W}^*, \boldsymbol{C}_{\vartheta})$ be the representation obtained from* (7) *with $\boldsymbol{W} = \boldsymbol{W}^*$, where $\boldsymbol{W}^*$ is obtained by solving* (5) *for a fixed $\boldsymbol{C}_{\vartheta}$. Under Assumptions 1–4, let $\boldsymbol{W}^*, \widetilde{\boldsymbol{W}}^*$ be the solutions of* (5) *obtained through PCA on fixed $\widetilde{\boldsymbol{C}}_V(0)$ and $\widetilde{\boldsymbol{C}}_V(t)$ respectively. Let λ_Z, λ_{Z+1} be Z^{th} and $(Z+1)^{th}$ eigenvalues of $\widetilde{\boldsymbol{C}}_V(0)$. Let $\zeta = 4\delta^{-1}\eta Q\sqrt{D}L_\phi^2 c_{in}^2$ and $\xi = 2^{\frac{7}{2}}\delta^{-1}QDc_{in}(L_\phi c_\theta + |\phi(0)|)$. There exists an orthogonal matrix $\boldsymbol{O}$ such that $\left\|\boldsymbol{O}^\top u(\boldsymbol{x}; \widetilde{\boldsymbol{W}}^*, \widetilde{\boldsymbol{C}}_V(t)) - u(\boldsymbol{x}; \boldsymbol{W}^*, \widetilde{\boldsymbol{C}}_V(0))\right\| \leq \zeta \frac{t}{\sqrt{M}}\left(1 + \frac{\xi \log M}{\lambda_Z - \lambda_{Z+1}}\right)$.*

Numerical Simulation on MNIST. Consider training (7) according to (10) under two settings: with $\widetilde{\boldsymbol{C}}_V(t)$ as a trainable matrix and with $\widetilde{\boldsymbol{C}}_V(0)$ as a fixed matrix. We look at the fractional difference between the learned representations after training in the two settings. In Figure 1 (c), the mean fractional difference of learned representations computed across samples is plotted. As expected, the difference goes down to zero as width increases.

Further Discussion and Open Problems

Our paper is the first to connect contrastive learning and NTK through the learning dynamics of non-linear neural networks, and would lead to further systematic study of trained contrastive models. We also note that while our results are not till convergence, they are still applicable for early stopped models which are often used in practice for contrastive learning. We make some final observations with regards to initialization and deeper networks as well as outline open problems arising from our derived results.

Effect of initialization on dimension collapse: For the presented analysis, we only assume a Gaussian initialization of weights (Lemmas 5 and 6 then hold with high probability). While this assumption is sufficient for proving the constancy of NTK results above, additional assumptions may be needed to obtain *meaningful* representations. While learning representations, it is desirable to avoid *dimension collapse*. Dimension collapse, in the context of linear contrastive models, has been shown for dot product (Esser, Mukherjee, and Ghoshdastidar 2023). Using NTK, we show that certain initialization schemes can cause dimension collapse even if cosine similarity based losses are used:

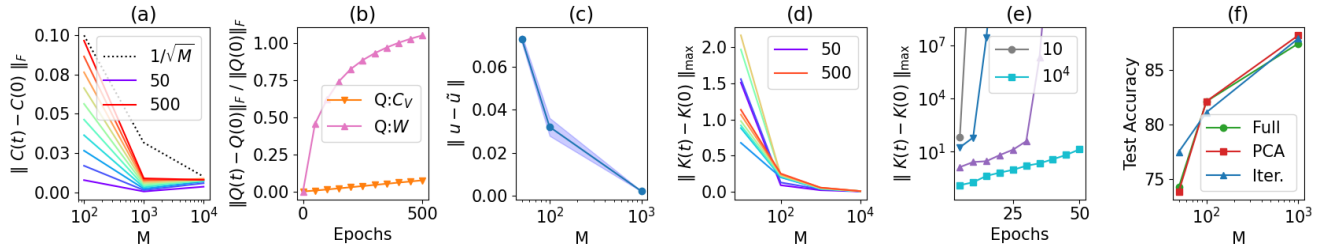

Figure 1: **(a)** Change in $\boldsymbol{C}_V$ during training for varying depths M. Time is indicated by color. **(b)** Evolutionof $\boldsymbol{C}_V$ and $\boldsymbol{W}$. Plotted are $\frac{\|\boldsymbol{C}_V(t)-\boldsymbol{C}_V(0)\|_F}{\|\boldsymbol{C}_V(0)\|_F}$ and $\frac{\|\boldsymbol{W}(t)-\boldsymbol{W}(0)\|_F}{\|\boldsymbol{W}(0)\|_F}$. **(c)** Difference in output when $\boldsymbol{C}_V$ is frozen and when $\boldsymbol{C}_V$ is trained for varying widths M. **(d)** Change in empirical NTK with *cosine similarity* for 3 layer networks of varying widths (time indicated by color). **(e)** Maximum entry-wise change in empirical NTK with *dot product similarity* for 3 layer networks of varying widths. **(f)** Accuracy on downstream task for PCA on $\widetilde{\boldsymbol{C}}(0)$, fully trained model and proposed iterative algorithm over 5 updates.

Proposition 18. *Consider a neural network of the form (2) trained using a loss of the form (1) using gradient descent. Then for any input* $\boldsymbol{x}$, $\boldsymbol{W}_{.i}(0) = \boldsymbol{W}_{.j}(0) \Rightarrow f_i(\boldsymbol{x};\theta(t)) = f_j(\boldsymbol{x};\theta(t)),\ \forall\, t \geq 0$.

Thus, collapse occurs irrespective of whether the NTK remains constant or not, and hence, this is applicable irrespective of the width of the neural network. Further, the result holds for both dot product and cosine similarity based losses. The result also holds for analytical NTK $\boldsymbol{K}^*$; if $\boldsymbol{K}^*$ is used in the dynamics in Lemma 9, then $f_i(\boldsymbol{x};\theta(0)) = f_j(\boldsymbol{x};\theta(0)) \Rightarrow f_i(\boldsymbol{x};\theta(t)) = f_j(\boldsymbol{x};\theta(t)),\ \forall t \geq 0$.

Empirical observations beyond the theory: **(i)** *Deep networks:* While Theorem 13 has been shown to hold only in the case of neural networks with a single hidden layer, we expect it to hold for deep networks as well. We experimentally examine the case with 3 hidden layers in Figure 1 (d). The results are similar to the case of a single hidden layer, as we again observe a decay with width that is roughly $\frac{1}{\sqrt{M}}$ up to scaling. Along similar lines, we observe in Figure 1 (e) that the NTK for three hidden layer networks optimized with dot product based losses diverges, similar to the single hidden layer case. **(ii)** *Iterative learning of the trace maximization problem:* Combining the findings from the previous section, we propose an alternative optimization procedure to solve (5) under $\boldsymbol{A} := \widetilde{\boldsymbol{C}}_V(t)$. As $\boldsymbol{C}_V$ updates slower than $\boldsymbol{W}$ as shown in Figure 1 (b), and since for a fixed $\widetilde{\boldsymbol{C}}_V$ the optimal $\boldsymbol{W}$ can be obtained using PCA, an alternative update could be by iteratively (i) updating $\boldsymbol{W}$ by solving PCA on $\widetilde{\boldsymbol{C}}_V(t-1)$ and (ii) updating $\boldsymbol{V}$ by running one step of gradient descent. We validate this approach in the following. **(iii)** *Predictive accuracy in downstream tasks:* Extending the results from the previous section, we show in Figure 1 (f) that the representations obtained with PCA on $\widetilde{\boldsymbol{C}}_V(0)$ and fully training (7) under (6) are close to each other with regards to downstream test accuracy for a simple linear classifier trained on the representations. While the previous section provides results on the behaviour of the PCA for different time-steps of $\widetilde{\boldsymbol{C}}_V$, Figure 1 (f) suggests that this similarity extends to PCA on $\widetilde{\boldsymbol{C}}_V(0)$ and fully trained networks as well. In addition, we also observe that the iterative optimization (as proposed above) performs well, especially for smaller widths (possibly due to the larger change in $\widetilde{\boldsymbol{C}}_V$ for small M).

Open problem (Missing link in the claim contrastive models perform PCA): In Lemma 17, we compare the solutions obtained by PCA on $\widetilde{\boldsymbol{C}}_V(0)$ and $\widetilde{\boldsymbol{C}}_V(t)$. But in Figure 1 (b), we observe that $\boldsymbol{C}_V$ and $\boldsymbol{W}$ evolve simultaneously, even though at different rates. The open problem then, is the exact connection between *fully trained contrastive models and the PCA solution.* We believe that a direct analysis, such as the one considered for Theorem 13, may not work. This is because the solutions might differ significantly at convergence even for closely initialized models if the eigengap is not significant. Therefore, a spectral viewpoint combined with the analysis of gradient descent steps is necessary to bound the deviation.

Open problem (NTK and PCA at convergence): Theorem 13 shows that, for wide networks, the NTK remains nearly constant for $M^{1/6}$ time-steps (in fact, one can also show constancy till $O(M^\alpha)$ steps for $\alpha < \frac{1}{4}$). While such results are in line with initial NTK analysis in the supervised setting (e.g. (Jacot, Gabriel, and Hongler 2018)), *it is an open question whether the constancy of NTK holds until convergence*. A potential approach to prove constancy till convergence would be to investigate the stationary points associated with cosine similarity based losses and verify if they are attained within M^α steps. While this is a crucial open question, we believe that the presented results provide the first valuable insights into the constancy of NTK under contrastive losses, beyond the squared error.

In Lemma 17, we show that outputs of the two considered trace maximization problems are close for $O(t(\log M)/\sqrt{M})$ steps. As the expression is in terms of time-steps, *the question remains if they are still close at convergence*. While we do not have a precise characterization of such results, a potential approach is to extend (Xu and Li 2021), who show that for $Z = 1$, PCA converges in roughly $O(\log(M))$ steps under Riemannian gradient descent.

Acknowledgements

The work was done in part when G. G. Anil was at TU Munich, supported by the German Academic Exchange Service (DAAD) through the DAAD KOSPIE fellowship, 2023. This work is also supported by the German Research Foundation (DFG) through the Priority Program SPP 2298 (project GH 257/2-1).

References

Arora, S.; Du, S. S.; Hu, W.; Li, Z.; Salakhutdinov, R. R.; and Wang, R. 2019a. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*.

Arora, S.; Khandeparkar, H.; Khodak, M.; Plevrakis, O.; and Saunshi, N. 2019b. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. In *International Conference on Machine Learning*.

Assran, M.; Caron, M.; Misra, I.; Bojanowski, P.; Bordes, F.; Vincent, P.; Joulin, A.; Rabbat, M.; and Ballas, N. 2022. Masked siamese networks for label-efficient learning. In *European Conference on Computer Vision*. Springer.

Balestriero, R.; and LeCun, Y. 2022. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *Advances in Neural Information Processing Systems*.

Bao, H.; Nagano, Y.; and Nozawa, K. 2022. On the Surrogate Gap between Contrastive and Supervised Losses. In *International Conference on Machine Learning*.

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

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*.

Cabannes, V.; Kiani, B.; Balestriero, R.; LeCun, Y.; and Bietti, A. 2023. The ssl interplay: Augmentations, inductive bias, and generalization. In *International Conference on Machine Learning*. PMLR.

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

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Chizat, L.; Oyallon, E.; and Bach, F. 2019. On lazy training in differentiable programming. *Advances in neural information processing systems*.

Edelman, A.; Arias, T. A.; and Smith, S. T. 1998. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*.

Esser, P.; Fleissner, M.; and Ghoshdastidar, D. 2023. Non-Parametric Representation Learning with Kernels. *arXiv preprint arXiv:2309.02028*.

Esser, P.; Mukherjee, S.; and Ghoshdastidar, D. 2023. Representation Learning Dynamics of Self-Supervised Models. *arXiv preprint arXiv:2309.02011*.

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. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*.

Han, L.; Ye, H.; and Zhan, D. 2023. Augmentation Component Analysis: Modeling Similarity via the Augmentation Overlaps. In *The Eleventh International Conference on Learning Representations*.

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

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*.

Ji, W.; Deng, Z.; Nakada, R.; Zou, J.; and Zhang, L. 2023. The power of contrast for feature learning: A theoretical analysis. *Journal of Machine Learning Research*.

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

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

Liu, C.; Zhu, L.; and Belkin, M. 2020a. On the linearity of large non-linear models: when and why the tangent kernel is constant. *Advances in Neural Information Processing Systems*.

Liu, C.; Zhu, L.; and Belkin, M. 2020b. Toward a theory of optimization for over-parameterized systems of non-linear equations: the lessons of deep learning. *arXiv preprint arXiv:2003.00307v1*.

Munkhoeva, M.; and Oseledets, I. 2023. Neural Harmonics: Bridging Spectral Embedding and Matrix Completion in Self-Supervised Learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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

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

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

Saunshi, N.; Plevrakis, O.; Arora, S.; Khodak, M.; and Khandeparkar, H. 2019. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*. PMLR.

Schölkopf, B.; Smola, A.; and Müller, K.-R. 1997. Kernel principal component analysis. In *International conference on artificial neural networks*. Springer.

Shah, A.; Sra, S.; Chellappa, R.; and Cherian, A. 2022. Max-Margin Contrastive Learning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence,*. AAAI Press.

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

Tian, Y. 2022. Understanding deep contrastive learning via coordinate-wise optimization. *Advances in Neural Information Processing Systems*.

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

Tosh, C.; Krishnamurthy, A.; and Hsu, D. 2021. Contrastive estimation reveals topic posterior information to linear models. *The Journal of Machine Learning Research*.

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

Wang, F.; and Liu, H. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Wang, R.; Chen, D.; Wu, Z.; Chen, Y.; Dai, X.; Liu, M.; Yuan, L.; and Jiang, Y.-G. 2023. Masked video distillation: Rethinking masked feature modeling for self-supervised video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

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

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

Wei, C.; Xie, S. M.; and Ma, T. 2021. Why Do Pretrained Language Models Help in Downstream Tasks? An Analysis of Head and Prompt Tuning. In *Advances in Neural Information Processing Systems*.

Xu, Z.; and Li, P. 2021. A Comprehensively Tight Analysis of Gradient Descent for PCA. *Advances in Neural Information Processing Systems*.

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

Zhuo, Z.; Wang, Y.; Ma, J.; and Wang, Y. 2023. Towards a Unified Theoretical Understanding of Non-contrastive Learning via Rank Differential Mechanism. In *The Eleventh International Conference on Learning Representations*.

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