

# Theoretical Foundations of Representation Learning using Unlabeled Data: Statistics and Optimization

Pascal Mattia Esser\*, Maximilian Fleissner†, Debarghya Ghoshdastidar‡

September 24, 2025

**Abstract.** Representation learning from unlabeled data has been extensively studied in statistics, data science and signal processing with a rich literature on techniques for dimension reduction, compression, multi-dimensional scaling among others. However, current deep learning models use new principles for unsupervised representation learning that cannot be easily analyzed using classical theories. For example, visual foundation models have found tremendous success using self-supervision or denoising/masked autoencoders, which effectively learn representations from massive amounts of unlabeled data. However, it remains difficult to characterize the representations learned by these models and to explain why they perform well for diverse prediction tasks or show emergent behavior. To answer these questions, one needs to combine mathematical tools from statistics and optimization. This paper provides an overview of recent theoretical advances in representation learning from unlabeled data and mentions our contributions in this direction.

## 1 Representation Learning: Past and Present

Over the past decade, there has been a paradigm shift in learning representations from unlabeled data, where the focus has shifted from data compression to learning Euclidean representations of (potentially) unstructured data. Unsurprisingly, the evolution of the representation learning problem is aligned with the increasing focus on visual and text data, specifically the growing reliance on large language models and visual foundation models to tackle diverse applications in computational data science and beyond. This paper discusses the theories of representation learning that are relevant in the current age of foundation models.

Before discussing recent theoretical advances, it is helpful to reflect on the key difference between the classical and modern representation learning paradigms. One may identify

---

\*Ludwig-Maximilians-Universität München, Department of Mathematics, Akademiestraße 7, 80799 München, email: [pascal.esser@math.lmu.de](mailto:pascal.esser@math.lmu.de)

†Technical University of Munich, TUM School of Computation, Information and Technology, Boltzmannstr. 3, 85748 Garching, email: [fleissner@cit.tum.de](mailto:fleissner@cit.tum.de)

‡Technical University of Munich, TUM School of Computation, Information and Technology, Munich Data Science Institute (MDSI), Munich Center for Machine Learning (MCML), Boltzmannstr. 3, 85748 Garching, email: [ghoshdas@cit.tum.de](mailto:ghoshdas@cit.tum.de)

two critical differences: (i) the prevalence of deep learning in current approaches towards representation learning and (ii) the context for learning representations of unlabeled data. We elaborate further on both aspects with a focus on how they impact the development of a statistical theory for modern (unsupervised) representation learning.

### 1.1 The importance of optimization in representation learning

An introductory course in multivariate statistics often covers a range of textbook methods for unsupervised representation learning, including principal component analysis (PCA), independent component analysis, factor analysis, projection pursuit, non-negative matrix factorization, among others [Mur22; Ize08; FT74]. Despite their prevalence in practice, a typical criticism of these approaches is that the learned representations are linear projections of the unlabeled data. There have been several attempts to generalize the principles of the aforementioned approaches to learn nonlinear representations. Notable classes of such methods are multidimensional scaling, including variants such as t-SNE [TSL00; Ize08; VH08], kernel methods [SSM98], tensor factorization [KB09], and neural network-based approaches such as self-organizing maps, restricted Boltzmann machines (RBM), or autoencoders [GBC16; Mur22].

The popularity of deep neural networks in the early 2010s led to the dominance of deep learning models for unsupervised representation learning [BCV13]. The shift in the research landscape is also influenced by the focus on image and text data, where it is known that unsupervised deep neural networks, such as RBMs, learn a hierarchy of visual features [Lee+11]. Deep learning also provides practical convenience:

**Why is deep representation learning so prevalent?** Unlike multivariate statistics principles, which often diverge in their practical implementations, current deep learning practices use a standardized framework to implement diverse ideas towards unsupervised representation learning. New approaches can be easily developed by applying standard optimization packages on a *new loss function* and a *new neural architecture*.

The unified implementation framework, compounded with the availability of more computational resources, has made deep learning the de facto model for representation learning, even beyond text or image domains—such as tabular data analysis [AP21; Yoo+20], where matrix-based methods are historically preferred.

Despite empirical success, deep representation learning has critical limitations. From a theoretical perspective, the key concern is the lack of rigorous understanding of the representations learned by the neural network-based models or the statistical properties of the trained models. An incomplete theory has severe practical implications. For example, design choices are often made based on empirical scaling laws [Hof+22] that can be inaccurate, learned representations are inherently neither transparent nor interpretable [Bom+21; Fu+24], and the models are prone to adversarial attacks [Hen+21].

The challenges in developing a theoretical foundation for deep learning are not restricted to the context of representation learning. It is widely acknowledged that classical theo-

ries of statistical generalization cannot explain the performance of supervised deep neural networks [Zha+21]. The principal bottleneck in providing a rigorous statistical understanding of deep learning lies in the complexity of the loss landscape of neural networks and the difficulty in characterizing the model learned via optimization. A key focus of recent theoretical research has been the integration of the optimization and statistical aspects of deep learning to provide a better characterization of the generalization properties of trained neural networks [Bel21].

### Open questions on optimization for deep representation learning.

Since the 1990s, there have been efforts to precisely characterize the dynamics of gradient descent in two-layer linear neural networks and to identify critical points of the dynamical system [BH89; Fuk98]. However, research in the past decade has provided a more precise description of the training dynamics of deep linear networks, typically in supervised settings, and the implicit biases of gradient-based optimization [SMG14; Sou+18]. The optimization dynamics remains complex for neural networks with non-linearities, although it is possible to identify the dynamics for infinitely wide networks, which closely reflects the learning dynamics of kernel machines under infinitesimally small step sizes of gradient descent—this is referred to as the *neural tangent kernel regime* or *lazy training regime* [CB20; JHG18; Aro+19; LZB20]. Recent works on tensor programs [YH21] and dynamic mean-field theory [BAP24; BCP24] provide a more accurate description of asymptotic dynamics under larger step sizes. However, existing theoretical studies on the optimization of deep neural networks focus mainly on supervised learning settings, specifically involving squared, logistic, or hinge losses [LZB20; CB20; Che+21]. While few works consider general loss functions [YH21; YL21; Sou+18], the analysis is restricted to only a few optimization steps or assuming strong data assumptions, such as data separability. This is a major gap in the theory of optimization in the context of learning representations from unlabeled data.

**Open questions.** What are the dynamics and implicit biases induced by optimization in the context of unsupervised representation learning? Specifically, how are the learning dynamics influenced by design choices common in representation learning, including architectures (bottleneck layers in autoencoders), type of loss functions (joint embedding loss used in self-supervised foundation models), nature of unlabeled data (masking or other augmentation), etc.?

The present paper addresses some of these questions, but with a special focus on their impacts on statistical properties.

### Open questions on statistical properties of trained neural networks

The fundamental techniques of learning theory, such as universal approximation or uniform convergence bounds [DHP21; Bel21], typically provide conservative guarantees for the generalization error of trained models and do not capture empirical trends [Zha+21]. Precise generalization error curves for simple supervised models have been studied since the late 1990s [SH02], but work in the last decade has accurately identified the connections

between training and generalization of learned models. For example, the bias of gradient descent towards small norm solutions causes the *double descent phenomenon* in over-parameterized models [Bel21; MM22] or in the presence of early stopping [HY21]. Similarly, implicit regularization induced by dimension reduction or optimization (Landweber iterations) achieves minimax optimal generalization error rates [DFH17].

To this end, the neural tangent kernel regime provides the most generally applicable technique for analytically obtaining exact generalization error curves for trained neural networks [MM22; BCP20], associated neural scaling laws [Bah+24], and robustness of models [BKM23; Sab+25]. However, kernel approximations are inaccurate for feedforward neural networks with finite-width and transformer/attention-based architectures. In such cases, more precise results on the generalization error can be obtained by accounting for the training of the hidden layers or the attention layers, typically known as *feature learning* and *attention learning*, respectively. This analysis is currently restricted to highly specialized supervised learning problems such as regression, Gaussian mixture classification, and sparse problems [Kar+21; Fu+23; SWL23; Ba+22]. Similar analysis of generalization in unsupervised deep learning is quite limited, and key questions remain unanswered.

**Open questions.** What are the statistical properties of unsupervised deep learning models? Are there provable benefits of deeper networks and attention mechanism over kernel methods in the context of learning representations from unlabeled data?

Preliminary steps towards answering these questions are discussed later in this paper in the context of reconstruction-based and joint embedding-based techniques for representation learning. However, a characterization of the generalization error in unsupervised representation learning, or generally the statistical performance of such models, is not possible without considering the practical relevance of such models. Next, we briefly review the context for modern deep representation learning models.

## 1.2 The influence of unsupervised representations on predictions

Traditionally, learning representations of unlabeled data are either for the purpose of data compression, which requires dimension reduction or matrix factorization techniques, or for visualization and exploratory data analysis, encompassing approaches for data clustering, multidimensional scaling, etc. Hence, the corresponding statistical theory often focuses on questions related to the error in estimating the latent signal, clusters, etc., or the loss of information [Gir21]. Importantly, prediction and generalization are not typical concerns in traditional unsupervised learning.

The objective of unsupervised representation learning has evolved in the current age of foundation models, where one relies on huge amounts of unlabeled data to learn “useful representation” that can be used in prediction tasks. In this context, representation learning is used to potentially reduce the dependence on expensive data labeling processes. For example, in large language models, where one uses next token prediction or masked modeling to learn useful embedding of unlabeled text data [Dev+19; Ope23]. Similarly, it is known that the representations learned in visual foundation models are naturally

suitable for object detection or image classification, even though representation learning does not use image labels [Car+21; BPL22].

Before proceeding to develop a statistical theory for modern representation learning, it is important to understand which aspects such a theory should address. To this end, we note two interesting concepts that have been popularized in current practice.

- **Self-supervision**

To learn useful representations from unlabeled data, it has become a common strategy to encode domain-specific knowledge through data augmentation. For example, the semantic meaning of an image does not change with rotations, and one learns representations that encode invariance to data augmentation [Bro+93]. The practice of learning representations from unlabeled data in conjunction with domain-specific augmentations is broadly referred to as *self-supervised learning*, which encompasses principles such as masked modeling [Dev+19], denoising-based techniques [Vin+10a], joint embedding approaches [BPL22], contrastive learning [Che+20], etc.

- **Emergent property**

The use of large-scale data is critical to the success of foundation models. It has been empirically observed that the predictive performance of both language and visual foundation models drastically improves when a large amount of unlabeled data is used for self-supervised representation learning. This phenomenon is often referred to as *emergent property* of foundation models [Bom+21; Car+21]. Figure 1 illustrates that the emergent property is an inherent characteristic of self-supervised representation learning and occurs even in simple examples.

In view of the above discussion, one can pose the following concrete theoretical questions on representation learning in foundation models.

**Open questions.** How can one derive guarantees on predictive performance of models that rely on self-supervised representation learning? What representations can be learned from the augmented data? What are the inductive biases of such models? Can one theoretically characterize emergent properties?

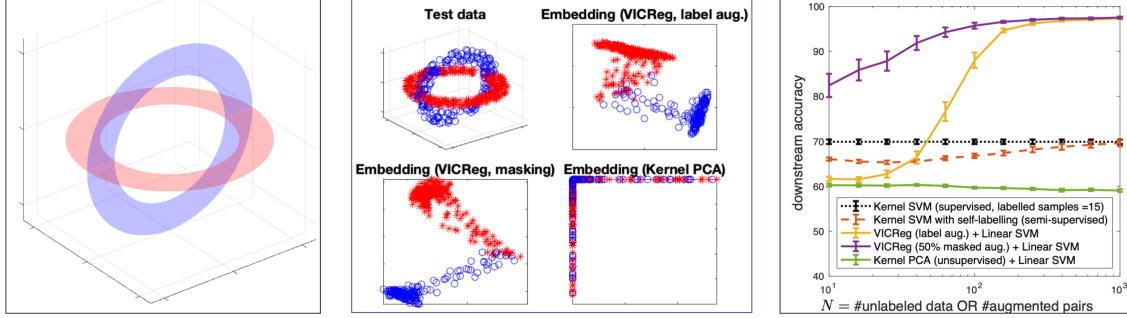


Figure 1: **Illustration of emergent properties of self-supervised representation learning in a simple setting.** The left plot shows the manifold of an example in  $\mathbb{R}^3$ , where we assume that data is generated from an union 2 classes (two intersecting 2-dimensional discs). Assume that one has access to 15 labeled samples and  $N$  unlabeled samples with  $N$  varying from 10 to 1000. The goal is to learn an “informative representation” from the unlabeled data so that a simple linear classifier, trained using only 15 labeled examples, correctly predicts the class labels for new samples.

We consider the case where one learns representations in  $\mathbb{R}^2$  learned using the unlabeled  $N$  samples. One could use traditional unsupervised methods like kernel principal component analysis (kernel PCA) [SSM98], or self-supervised techniques. We specifically consider a joint-embedding approach VICReg [BPL22; Cab+23], where for each unlabeled sample  $x \in \mathbb{R}^3$ , one generates a random augmented view  $x^+$  either by *masking* each coordinate of  $x$  with probability 0.5, or by *rotating* the sample within the disc that it lies in (the latter is a hypothetical *label-dependent augmentation* based on the philosophy that rotating images preserves their semantic meaning and the augmented data remains on same manifold).

The middle plot shows the embedding of 500 labeled test samples, where the representations  $f : \mathbb{R}^3 \rightarrow \mathbb{R}^2$  is learned with  $N = 1000$  unlabeled examples. The plot shows that  $f(\cdot)$  from VICReg, with either augmentation, almost separates the two classes, whereas kernel PCA learns an uninformative representation. The right plot shows the downstream predictive performance of unsupervised representation learning with varying  $N$ , averaged over 100 independent runs. The downstream classifier is a linear support vector machine (SVM) trained on the representation  $f(\cdot)$  of the 15 labeled samples. We also include two baselines that do not use representation learning—a supervised kernel SVM trained on the 15 labeled samples in  $\mathbb{R}^3$ , and a semi-supervised approach of self-labeling, where kernel SVM predicts on available  $N$  unlabeled data, and uses the pseudo-labels to update the model. Supervised kernel SVM provides a baseline of 70% accuracy, which does not improve with semi-supervised techniques. Unsupervised representation learning with kernel PCA does not learn “more informative” representations with more unlabeled data, whereas VICReg shows *emergent behavior*—the downstream classification performance increases with the availability of large amount of unlabeled, augmented data. The improvement is particularly insightful for VICReg with label-dependent augmentation, whose performance is similar to kernel PCA when there are few unlabeled samples, but significantly improves when more unlabeled data is used.

### 1.3 Focus of this Work

The broad aim of this paper is to discuss recent theoretical advances in characterizing the optimization and statistical properties of unsupervised and self-supervised representation learning, along with some discussion on their generalization and emergent properties. Specifically, various results are presented on the dynamics of the learned representation under gradient descent, characterization of the learned representation, and the generalization error of the downstream predictors. The technical results are presented considering two specific principles for representation learning, illustrated in Figure 2:

- **Reconstruction**

In this original form, a reconstruction-based model learns a potentially low-dimensional representation through the process of reconstructing unlabeled examples. Autoencoder architectures are commonly used in these contexts, and if the activation of the hidden layer is linear, the learned representation is closely related to PCA [BH89]. In the context of self-supervised settings, the data augmentation is in the form of introducing noise in unlabeled data or masking them, leading to *denoising autoencoders* [Vin+10a] and *masked autoencoders* [Dev+19], respectively.

- **Joint embedding**

In its basic form, joint embedding aims to learn a joint representation of unlabeled augmented samples, such that the representation is invariant under augmentations. Diverse formulations of this philosophy have been proposed, which can be broadly characterized into two principles. Non-contrastive learning aims to learn a representation that is identical for every augmented pair of samples [Bro+93; BPL22; Car+21], while contrastive learning additionally uses non-augmented pairs to ensure that a nontrivial representation is learned [Che+20].

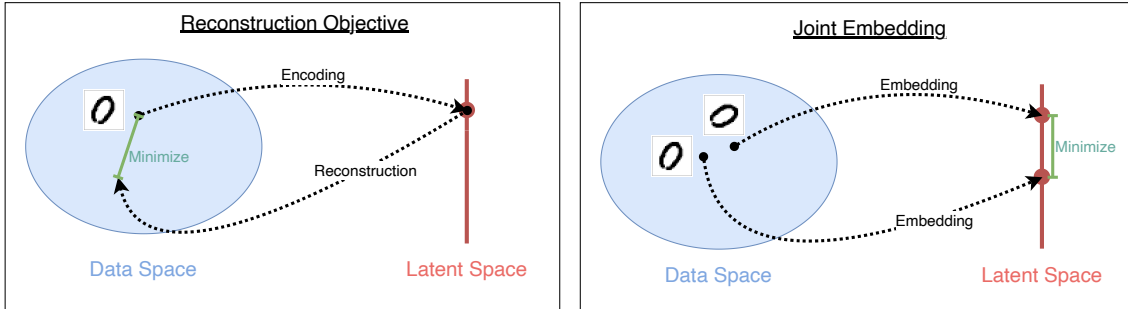


Figure 2: **Illustration of two principles for representation learning.** (Left) The objective of reconstruction. An data instance is mapped into a lower dimensional latent space using an *encoder function* and then mapped back to the original feature space using a *reconstruction function*. The functions are learned by minimizing the distance of the reconstruction from either the given instance or its augmentation. (Right) The objective of joint embedding. This principle builds on the idea that semantically similar pair of data instances, usually obtained through data augmentation, should be embedded close to each other in the latent space. Hence, the embedding function is learned by minimizing the distance between the embedding augmented pairs of data instances, incorporating additional measures to ensure that trivial embeddings are not learned.



Sections 2 and 3, respectively, focus on reconstruction-based approaches based on autoencoders, and joint embedding approaches, including contrastive and non-contrastive methods. Section 4 discusses some results on the generalization error bounds for downstream prediction tasks using learned representations. Finally, we provide an outlook and key future research directions in Section 5.

In the subsequent discussion, we use the following notation. We denote matrices by bold capital letters  $\mathbf{A}$ , vectors as  $\mathbf{a}$ , and  $\mathbf{I}_m$  for an identity matrix of size  $m \in \mathbb{N}$ . We assume that the  $N$  data points  $\{\mathbf{x}_1 \cdots \mathbf{x}_N\} \in \mathbb{R}^d$  are collected in a data matrix  $\mathbf{X} \in \mathbb{R}^{d \times N}$ . We use  $\|\cdot\|$  to denote the Euclidean or  $L_2$ -norm for vectors, the Frobenius norm for matrices, and the Hilbert-Schmidt norm for operators.  $O(\cdot)$ ,  $\Omega(\cdot)$ ,  $\Theta(\cdot)$  refers to asymptotic notation. Other notation is introduced in the respective sections.

## 2 Reconstruction-based methods using Autoencoders

Autoencoders (AE) [Kra91] are prototypical examples of neural networks used for reconstruction. Formally, an AE architecture is a composition of an encoder function  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$  followed by a decoder function  $g : \mathbb{R}^k \rightarrow \mathbb{R}^d$ , where  $\mathbb{R}^d$  is the original feature space and  $\mathbb{R}^k$  is the latent space. Traditionally, this setup has been used to learn to reconstruct data  $g(f(\mathbf{x})) \approx \mathbf{x}$ . Given an unlabeled data matrix  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ , the most common approach to learning the encoder and decoder is to minimize the reconstruction error.

$$\min_{f, g} \sum_{i=1}^N \|\mathbf{x}_i - g(f(\mathbf{x}_i))\|^2. \quad (1)$$

Alternative formulations are also used in practice, including minimizing energy functions [Ran+06] or imposing additional unsupervised tasks, such as clustering, on the learned representations [Yan+17]. If the dimension of the latent space or the *bottleneck layer*  $k < d$ , then the obtained representation  $f(\mathbf{x}) \in \mathbb{R}^k$  provides a *compression* of the data  $\mathbf{x} \in \mathbb{R}^d$ . The representations learned by a trained encoder  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$  have found success in a wide range of tasks, making AEs one of the popular deep learning architectures [BCM05; BCV13; Yan+17].

One of the fundamental theoretical questions is to characterize the representation  $\mathbf{x} \mapsto f(\mathbf{x})$  learned by optimization. To this end, it is useful to consider a linear AE with a single hidden layer (bottleneck), that is,  $f(\mathbf{x}) = \mathbf{W}_1 \mathbf{x}$  and  $g(f(\mathbf{x})) = \mathbf{W}_2 f(\mathbf{x})$ , where  $\mathbf{W}_1 \in \mathbb{R}^{k \times d}$  and  $\mathbf{W}_2 \in \mathbb{R}^{d \times k}$  are trainable weight matrices. Hence, one may replace Equation (1) by the following optimization:

$$\min_{\mathbf{W}_1, \mathbf{W}_2} \|\mathbf{X} - \mathbf{W}_2 \mathbf{W}_1 \mathbf{X}\|^2 + \lambda \cdot r(\mathbf{W}_1, \mathbf{W}_2) \quad (2)$$

where  $r$  denotes a regularization on the weight matrices, and  $\lambda > 0$  is a constant. The training dynamics and the critical points for linear AEs can be characterized. In particular, [BH89] show that linear AE without regularization is equivalent to PCA—it finds



solutions in the principal component that spans the subspace, but the individual components and the corresponding eigenvalues cannot be recovered. [Kun+19] show that standard  $l_2$  regularization reduces symmetry solutions to the group of orthogonal transformations. Finally, [Bao+20] show that nonuniform  $l_2$  regularization allows linear AE to recover ordered, axis-aligned principal components. Beyond linear AEs, [RG22] provides the learning dynamics when the encoder  $f$  is a nonlinear function.

**Does data compression lead to good representations?** Since the modern focus of the representation learning extends beyond data compression, it is natural to question whether reconstruction-based AEs (or nonlinear extensions of PCA) learn useful representations. Figure 3 illustrates examples where standard AEs are not suitable and alternative principles are needed.

In this paper, we will consider and theoretically analyze two alternatives to simple reconstruction losses of the form Equation (1).

### 1. Reconstruction objectives based on augmented data.

Current AE-based representation learning models go beyond mere input reconstruction and define self-supervised objectives using augmented data. The most prominent self-supervised tasks involve denoising [Vin+10a], where  $\mathbf{x}^+$  is obtained by adding random noise to  $\mathbf{x}$ , and masking [Dev+19; He+22], where parts of  $\mathbf{x}$  are removed or masked to construct  $\mathbf{x}^+$ . Subsequently, we present some theoretical

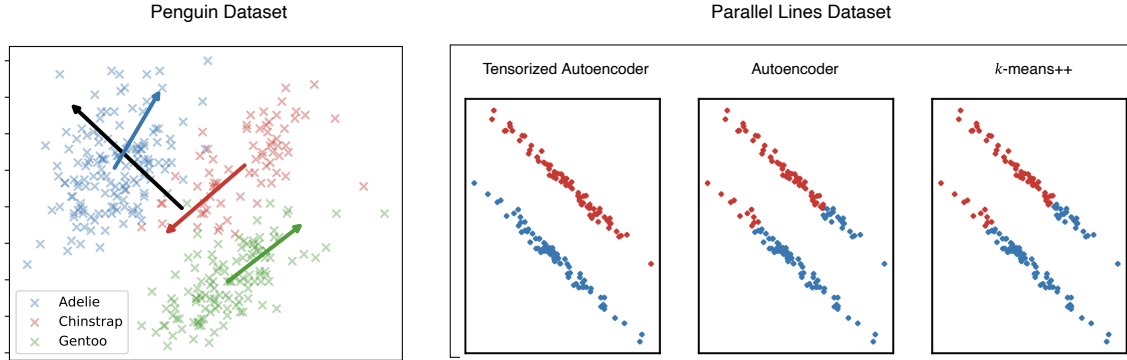


Figure 3: Source [Ess+23]. (Left) Illustration of the Simpson’s paradox [Sim51]. The scatter plot shows the dataset of 2-dimensional features for three different species of penguins [GWF14]. The three clusters, for the different species, and their first principal component are plotted in red, blue and green, respectively. A linear AE or PCA into  $k = 1$  dimensional latent space can only recover the principal component of the full dataset (shown in black), but cannot capture the characteristics of the individual species (clusters). Such examples of Simpson’s paradox can only be found in non-linear models and real-world applications such as social or science and medical science.

(Right) Performance of different clustering algorithms on Simpson’s paradox data. We consider a synthetic version of Simpson’s paradox with noisy samples from two parallel lines in  $\mathbb{R}^2$ . One can either apply  $k$ -means++ on the original data (right), or on learned representations  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ . AE recovers the principal component of the full data, which does align with the direction of the clusters. Hence, clustering the representations from AE does not recover the true clusters (middle). [Ess+23] introduce tensorized AE, which learns representations for individual clusters, and results in better clustering performance.

results for denoising autoencoders (DAEs) [BCM05], where the input data  $\mathbf{X}$  are corrupted by noise, which the model needs to ignore when reconstructing  $\mathbf{X}$ . In Section 2.1, we present an analysis of the learning dynamics and characterization of the generalization error of linear DAEs, proposed in [HFG25].

## 2. Generalized representations and new AE architecture.

An alternative approach to learning better representations is to generalize the notion of a representation. Unlike standard AE that learns a single encoding  $\mathbf{x} \mapsto f(\mathbf{x})$ , the idea is embed different inputs differently. In Section 2.2, we discuss a specific architecture of tensorized AE [Ess+23], where the bottleneck layer can learn multiple representations. Similar philosophy also lies behind deep clustering networks [Yan+17].

### 2.1 Learning dynamics and generalization error of denoising AEs

Denoising autoencoders (DAEs) are trained to denoise unlabeled input data, but it is widely acknowledged that the resulting trained encoder learns useful low-dimensional representations [Vin+10b], which leads to its popularity in visual data analysis. For the purpose of theoretical analysis, we restrict the discussion to linear DAEs with  $f(\mathbf{x}) = \mathbf{W}_1 \mathbf{x}$  and  $g(f(\mathbf{x})) = \mathbf{W}_2 f(\mathbf{x})$ , where the training objective is formulated as

$$\min_{\mathbf{W}_1, \mathbf{W}_2} \mathcal{L}_{\text{trn}}(\mathbf{W}_1, \mathbf{W}_2) = \|\mathbf{X} - \mathbf{W}_2 \mathbf{W}_1 (\mathbf{X} + \mathbf{A})\|^2 + \lambda \cdot \|\mathbf{W}_2 \mathbf{W}_1\|^2, \quad (3)$$

where  $\mathbf{A} \in \mathbb{R}^{d \times N}$  denotes a noise matrix and the regularization of the form  $r(\mathbf{W}_1, \mathbf{W}_2) = \|\mathbf{W}_2 \mathbf{W}_1\|^2$ . In (3), one could denote  $\mathbf{W}_* = \mathbf{W}_2 \mathbf{W}_1$  and optimize only over  $\mathbf{W}_*$ . This results in a *linear denoiser* [SN23; KSS24], which can be solved as multivariate ridge regression. Although the statistical properties of ridge regression have been well studied, typical results assume additive noise is in the output [Bel21; DFH17]. In contrast, a precise characterization of the generalization error of linear denoisers requires the additive noise to be taken into account. We refer to [SN23; KSS24] for the generalization error of linear denoisers, which exhibit interesting double-descent behavior in the over-parameterized regime.

In the subsequent discussion, we focus on the practically relevant setting where  $\mathbf{W}_1, \mathbf{W}_2$  are separately trained and the bottleneck layer is of dimension  $k < d$ . Studying this linear DAE with bottleneck allows us to precisely investigate the influence of the bottleneck dimension on the learned representations. We consider the over-parameterized (or high-dimensional) regime, where the number of input observations is dominated by the dimensionality of the features,  $c := \frac{d}{N} > 1$ . For ease of exposition, it is convenient to assume that  $\mathbf{X} + \mathbf{A}$  has full rank and the matrices  $\mathbf{X} \mathbf{X}^\top$  and  $\mathbf{A} \mathbf{A}^\top$  have simple eigenvalues. Under this setting, one can rely on the classical result of [BH89] on critical points for linear neural networks, which leads to the following result.

**Theorem 2.1** (Global minimizer of linear DAEs [HFG25]). *Consider Equation (3) in the ridgeless limit ( $\lambda \rightarrow 0$ ), and denote  $\mathbf{W}_* = \mathbf{W}_2 \mathbf{W}_1$ . The global minimizer  $\mathbf{W}_*$  of Equa-*

tion (3) converges to  $\mathbf{W}_* = P_{[k]}(\mathbf{X})(\mathbf{X} + \mathbf{A})^\dagger$ , where  $P_{[k]}(\mathbf{X})$  is the rank- $k$  approximation to  $\mathbf{X}$  and  $\dagger$  denotes the Moore-Penrose pseudoinverse.

It is important to note that the above solution is distinct from PCA, due to the noise  $\mathbf{A}$  inside the pseudoinverse, and also distinct from linear denoisers due to the rank- $k$  approximation. Using Theorem 2.1, it is possible to characterize the generalization error of DAEs, under mild assumptions about the data-generating process. For training, we consider features  $\mathbf{X} \in \mathbb{R}^{d \times N}$  and an additive Gaussian noise matrix  $\mathbf{A} \in \mathbb{R}^{d \times N}$  with entries sampled independently from  $\mathcal{N}\left(0, \frac{\eta_{\text{trn}}^2}{d}\right)$ . We study generalization in terms of the expected denoising error on test data

$$\mathcal{L}_{\text{tst}}(\mathbf{W}_*) := \frac{1}{N_{\text{tst}}} \mathbb{E}_{\mathbf{A}_{\text{trn}}, \mathbf{A}_{\text{tst}}} [\|\mathbf{X}_{\text{tst}} - \mathbf{W}_*(\mathbf{X}_{\text{tst}} + \mathbf{A}_{\text{tst}})\|^2], \quad (4)$$

$\mathbf{X}_{\text{tst}} \in \mathbb{R}^{d \times N_{\text{tst}}}$  is the test data, perturbed by Gaussian noise matrix  $\mathbf{A}_{\text{tst}} \in \mathbb{R}^{d \times N_{\text{tst}}}$ , that is, the entries of  $\mathbf{A}_{\text{tst}}$  are  $\mathcal{N}\left(0, \frac{\eta_{\text{tst}}^2}{d}\right)$  and  $\eta_{\text{trn}}, \eta_{\text{tst}} = \Theta(1)$ .

For the following results, assume that the training data  $\mathbf{X}$  and the test data  $\mathbf{X}_{\text{tst}}$  lie in the same low-dimensional subspace of dimension  $r \ll d, N$ . Formally, the test data  $\mathbf{X}_{\text{tst}}$  satisfy  $\mathbf{X}_{\text{tst}} = \mathbf{U}\mathbf{L}$ , for  $\mathbf{U} \in \mathbb{R}^{d \times r}$  the left singular vectors of  $\mathbf{X}$  and for some non-zero coefficient matrix  $\mathbf{L} \in \mathbb{R}^{r \times N_{\text{tst}}}$ . The low-rank assumption on the data is well supported by empirical evidence that real-world data sets are approximately low-rank, as argued in [UT19]. In addition, we assume that  $\|\mathbf{X}\|_2 = \Theta(1)$  and the ratio between the largest and the smallest nonzero singular values of  $\mathbf{X}$  is  $\Theta(1)$ . With these assumptions in place, we may state the following.

**Theorem 2.2** (Generalization error of over-parametrized linear DAEs [HFG25]). *Let  $\sigma_i$  denote the  $i$ -th singular value of  $\mathbf{X}$ , and define  $\alpha_i := \sigma_i \eta_{\text{trn}}^{-1}$ . Let  $d \geq N + r$ , and  $c := \frac{d}{N}$ . Let  $\mathbf{J} \in \mathbb{R}^{r \times r}$  be the diagonal matrix with  $\mathbf{J}_{ii} = (\alpha_i^2 + 1)^{-2} \cdot \mathbb{1}_{i \in [k]} + \mathbb{1}_{i \notin [k]}$ , where  $\mathbb{1}_{(\cdot)}$  denotes the indicator function. Then,*

$$\mathcal{L}_{\text{tst}}(\mathbf{W}_*) = \frac{1}{N_{\text{tst}}} \text{Tr}(\mathbf{J}\mathbf{L}\mathbf{L}^\top) + \frac{\eta_{\text{tst}}^2 c}{d(c-1)} \sum_{j \in [k]} \frac{\alpha_j^2}{1 + \alpha_j^2} + O\left(\frac{1}{d^2}\right).$$

Since  $\mathbf{L}$  depends only on the test data, the overall magnitude of  $\text{Tr}(\mathbf{J}\mathbf{L}\mathbf{L}^\top)$  is mainly influenced by the size of the diagonal entries of  $\mathbf{J}$ . Thus, the first term *decreases* as bottleneck dimension  $k$  increases toward  $r$ . In contrast, the second term *increases* as  $k$  grows. This trade-off behavior aligns exactly with the *classical understanding of the bias-variance trade-off*. In fact, one can show that the first term is exactly the bias of  $\mathbf{W}_*$  whereas the second is the variance of the predictor. In particular, the aforementioned trade-off emerges in the over-parameterized regime, where  $d > N$ . Figure 4 (left) shows the generalization error for linear DAE as a function of the ratio  $c = \frac{d}{N}$ . We observe the typical phenomenon that over-parameterization leads to a decrease in generalization error as well-known in the *double descent literature* [Bel21]. However, in the case of the DAE, the complexity of the model is controlled by the width of the bottleneck  $k$ , not only the input dimension.

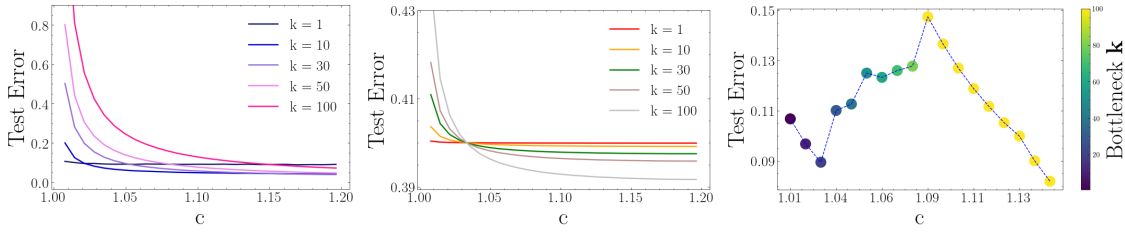


Figure 4: Source [HFG25]. **Impact of bottleneck dimension of linear DAE on generalization error.** We plot the test errors of linear DAEs with and without skip connection on CIFAR-10, illustrating how the bottleneck dimension  $k$  and  $c = \frac{d}{N}$  jointly influence generalization. For the experiments, each sample was reshaped into a 3072-dimensional vector, and the rank of the dataset was set to  $r = 100$  using SVD. Since the dataset has a fixed ambient dimension  $d$ , our numerical experiments focus on varying the number of training samples  $N$ . (Left & Middle) The left and middle plot show the denoising error on test data with varying  $c = \frac{d}{N}$  for the linear DAEs without and with skip connections, respectively. Both plots demonstrate that the optimal choice of  $k$  depends on the over-parameterization ratio  $c$ , reflecting a distinct bias-variance trade-off in different regimes. (Right) To study the impact of over-parameterization, the right plot is constructed by jointly increasing both  $k$  and  $c$  in the model without skip connections. In particular, this plot demonstrates that jointly increasing  $c$  and bottleneck dimension  $k$  leads to a *second peak* in the test curve within the overparameterized regime.

It is further useful to study a practical variant of DAE, where a skip connection is added across the bottleneck layer resulting in the following optimization.

$$\mathcal{L}_{\text{trn}}(\mathbf{W}_1, \mathbf{W}_2) = \|\mathbf{X} - (\mathbf{I} + \mathbf{W}_2 \mathbf{W}_1)(\mathbf{X} + \mathbf{A})\|^2 + \lambda \cdot \|\mathbf{W}_2 \mathbf{W}_1\|^2. \quad (5)$$

Here  $\mathbf{I}$  is the  $d$ -dimensional identity matrix that maps inputs directly to the output layer, without compressing them in the bottleneck. Although adding a skip connection is counterintuitive for AEs, practical implementations such as the U-Net architecture [RFB15] routinely incorporate skip connections as a core architectural feature. It is an open question to understand the impact of skip connection on the generalization error in DAEs. We extend Theorems 2.1–2.2 to address this question.

**Theorem 2.3** (Global minimizer and generalization error of linear DAE with Skip Connection [HFG25]). *In the ridgeless limit  $\lambda \rightarrow 0$ , the global minimizer  $\mathbf{W}_*$  of Equation (5) approaches  $\mathbf{W}_* = -P_{[k]}(\mathbf{A})(\mathbf{X} + \mathbf{A})^\dagger$ . Furthermore, under similar assumptions as Theorem 2.2, the generalization error  $\mathcal{L}_{\text{tst}}$  is given by*

$$\begin{aligned} \mathcal{L}_{\text{tst}}(\mathbf{W}_*) &= \eta_{\text{tst}}^2 \left(1 - \frac{k}{d}\right) + \frac{k}{dN_{\text{tst}}} \text{Tr}(\mathbf{J}^{\text{sc}} \mathbf{L} \mathbf{L}^\top) + \frac{\eta_{\text{tst}}^2 k}{d^2} \frac{c}{c-1} \sum_{i=1}^r \frac{\sigma_i^2}{(\eta_{\text{trn}}^2 + \sigma_i^2)} \\ &\quad + \frac{3\eta_{\text{tst}}^2 k}{dN} \frac{1}{c} \sum_{i=1}^r \frac{\eta_{\text{trn}}^2 \sigma_i^2}{(\eta_{\text{trn}}^2 + \sigma_i^2)} + O\left(\frac{1}{dN_{\text{tst}}}\right). \end{aligned}$$

where  $\mathbf{J}^{\text{sc}}$  is a diagonal matrix with  $J_{ii}^{\text{sc}} = \frac{c+(c-1)\sigma_i^2}{c(1+\eta_{\text{trn}}^2\sigma_i^2)^2}$  for each  $i \in [r]$ .

It is useful to compare Theorem 2.2 with Theorem 2.3 (also see Figure 4). Firstly, observe that the variance term in the former is responsible for the sharp increase in the test curves as the ratio  $c$  approaches 1, due to the term  $(c-1)^{-1}$ . A similar, but less pronounced trend

is observed in the model with skip connections, where the term  $\frac{\eta_{\text{test}}^2 k}{d^2} \frac{c}{c-1} \sum_{i=1}^r \frac{\sigma_i^2}{(\eta_{\text{trn}}^2 + \sigma_i^2)}$  also includes the factor  $(c-1)^{-1}$ . However, in contrast to the model without skip connections, the expression is multiplied with an additional factor of  $d^{-1}$ . This suggests that *skip connections help mitigate the sharp rise in variance* that typically occurs when the model is in the moderately over-parameterized regime, leading to more stable generalization performance.

## 2.2 Learning multiple representations with tensorized autoencoders

We now consider an example where the AE architecture is generalized to incorporate additional structures in the data. Conceptually, the broad aim is to learn representations that preserve important structures of the data, while removing noise dimensions. Linear AE or PCA retains only the high-variance directions, but it could sometimes be more relevant to preserve other topological properties. In this section, we restrict ourselves to the case where the goal is to learn latent cluster representations, which can also be used to cluster the data (see Figure 3).

In [Ess+23], we introduce a modified AE architecture that we term *Tensorized Autoencoders (TAE)*. At a high level, the TAE architecture consists of  $m$  AEs in parallel. Given a data set  $\{\mathbf{x}_i\}_{i=1}^N$ , each input  $\mathbf{x}_i$  is passed to  $j^{\text{th}}$  AE with a nonnegative weight  $\mathbf{S}_{j,i}$ . This allows us to learn  $m$  different latent representations, one from each AE.

Assuming that a data set has  $m$  linearly separable clusters, one can show that tensorized linear AEs provably recover the principal directions of *each cluster* while jointly learning the assignment of clusters. To see this, we define two-layer linear TAEs formally as follows. For each data point  $\mathbf{x}_i$ , define  $\tilde{\mathbf{x}}_i^{(j)} := \mathbf{x}_i - \mathbf{c}_j$  as a centered data, where  $\mathbf{c}_1, \dots, \mathbf{c}_m$  are cluster centers learned defined later. The  $j$ -th AE in the TAE is parameterized by the linear encoder map,  $\tilde{\mathbf{x}}_i^{(j)} \mapsto \mathbf{W}_1^{(j)} \tilde{\mathbf{x}}_i^{(j)}$ , where  $\mathbf{W}_1^{(j)} \in \mathbb{R}^{k \times d}$  (we view the encoder as the embedding of the  $j$ -th cluster). Similarly, the linear decoder map for  $j$ -th AE is parameterized by  $\mathbf{W}_2^{(j)} \in \mathbb{R}^{d \times k}$ . The TAE parameters are learned by solving the following optimization.

$$\begin{aligned} \min_{\{\mathbf{W}_2^{(j)}, \mathbf{W}_1^{(j)}\}_{j=1}^m, \mathbf{S}} \quad & \sum_{i=1}^N \sum_{j=1}^m \mathbf{S}_{j,i} \left[ \left\| \tilde{\mathbf{x}}_i^{(j)} - \mathbf{W}_2^{(j)} \mathbf{W}_1^{(j)} \tilde{\mathbf{x}}_i^{(j)} \right\|^2 + \lambda \left\| \mathbf{W}_1^{(j)} \tilde{\mathbf{x}}_i^{(j)} \right\|^2 \right], \\ \text{s.t.} \quad & \sum_{j=1}^m \mathbf{S}_{j,i} = 1, \quad \mathbf{S}_{j,i} \geq 0, \quad \mathbf{W}_1^{(j)} \mathbf{W}_1^{(j)T} = \mathbf{I}_k, \quad \mathbf{c}_j = \frac{\sum_{i=1}^N \mathbf{S}_{j,i} \mathbf{x}_i}{\sum_{i=1}^N \mathbf{S}_{j,i}}, \end{aligned} \quad (6)$$

where  $\lambda > 0$  is a regularization constant. Interpreting  $\mathbf{S}_{j,i}$  as the probability that  $\mathbf{x}_i$  is from the  $j$ -th cluster, the above objective can be interpreted as minimizing the expected reconstruction error of the samples, regularized with a  $k$ -means clustering cost. The constraints on  $\{\mathbf{S}_{j,i}\}_{j=1}^m$  ensure that the weights are indeed probabilities and the orthogonality of  $\mathbf{W}_1^{(j)}$  ensures that the encoders are projections. The following theorem

characterizes the optimal parameters of a linear TAE, showing that linear TAEs indeed learn clustering-specific representations of the data.

**Theorem 2.4** (Parameterization at optimal for TAE [Ess+23]). *For  $0 < \lambda \leq 1$ , optimizing Equation 6 results in the parameters at the optimum satisfying the following:*

1. *Class Assignment: Any optimal  $\mathbf{S}$  satisfies  $\mathbf{S}_{j,i} \in \{0, 1\}$ . In combination with the linear constraint, the above implies that each  $\mathbf{x}_i$  is assigned exactly to one cluster and is encoded by exactly one AE.*
2. *Encoding / Decoding (learned weights): At optimality,  $\mathbf{W}_2^{(j)T} = \mathbf{W}_1^{(j)}$  correspond to the top  $k$  eigenvectors of  $\hat{\Sigma}_j := \sum_{i=1}^N \mathbf{S}_{j,i} (\mathbf{x}_i - \mathbf{c}_j) (\mathbf{x}_i - \mathbf{c}_j)^\top$ .*

As a consequence, for any optimal solution,  $\mathbf{c}_j$  and  $\hat{\Sigma}_j$  act as estimates for the means and covariances for each specific class, respectively.

The above theorem demonstrates the suitability of the TAE formulation. In practice, it is more useful to consider nonlinear AEs,  $f_j(\cdot), g_j(\cdot)$ , resulting in an optimization of the form:

$$\min_{\{f_j, g_j\}_{j=1}^m, \mathbf{S}} \sum_{i=1}^N \sum_{j=1}^m \mathbf{S}_{j,i} \left[ \left\| \tilde{\mathbf{x}}_i^{(j)} - g_j \left( f_j \left( \tilde{\mathbf{x}}_i^{+(j)} \right) \right) \right\| + \lambda \cdot r(f_j, g_j) \right],$$

where it is implicitly assumed that optimization is performed over a certain parametric form of the encoder  $f_j$  and decoder  $g_j$ . General regularization functions can be imposed on the AEs, but regularization based on  $k$ -means is popular in the deep clustering literature [Yan+17]. Further empirical work also extends the idea of tensorization beyond AEs to tensorized variational AEs and restricted Boltzmann machines [SAE24]. However, it is challenging to characterize the optimal solution or the training dynamics beyond linear TAEs.

### 3 Self-supervised joint embedding methods

AE architecture and its variants focus on reconstructing or recovering the original data  $\mathbf{x}$ . As a consequence, the learned representation  $\mathbf{x} \mapsto f(\mathbf{x})$  ignores the data features that have less impact on reconstruction. This may not be desirable if the representations are used for downstream prediction tasks. One can see this in the following simple example.

**Example (AE may ignore direction relevant for Gaussian mixture classification)** Consider data sampled from Gaussian mixture distribution in  $\mathbb{R}^2$

$$\mathbf{x} = \begin{pmatrix} x^{(1)} \\ x^{(2)} \end{pmatrix} \sim \frac{1}{2} \mathcal{N} \left( \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \right) + \frac{1}{2} \mathcal{N} \left( \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \right).$$

If a linear AE is used with a bottleneck layer of width 1, then asymptotically as  $n \rightarrow \infty$ , the AE learns the principal component  $\mathbf{x} \mapsto f(\mathbf{x}) = x^{(2)}$ . However,

if the objective is to use the representations for downstream classification, it is desirable to project the data on the axis that separates the classes,  $\mathbf{x} \mapsto x^{(1)}$ .

The above situation, where the useful features do not align with the features learned by reconstruction, is often encountered in visual foundation models [BL24]. In these cases, *joint embedding methods* are preferred. These methods rely on pairs  $\mathbf{x}, \mathbf{x}^+ \in \mathbb{R}^d$  (or tuples) of semantically similar samples and learn a representation  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$  such that  $f(\mathbf{x}), f(\mathbf{x}^+)$  are “close” or “aligned”. The idea was introduced in [Bro+93], where a Siamese network—two neural networks with identical weights—was used on pairs of signatures from the same person for the application of signature verification. In practice, semantically similar samples are generated through *data augmentation*, which depends on the data domain. For example, image augmentation through random cropping, rotation, color jitter, etc. help to learn useful representations for image classification [Che+20]. In current foundation models, labeled data are not used to learn the representation (often called *self-supervised pretraining*). The hope is that the augmented pair  $\mathbf{x}, \mathbf{x}^+$  belong to the same class for downstream prediction tasks.

**Examples (Data augmentation resulting in same or different class labels)** In the context of classifying cat images from dog images, augmentation such as cropping or rotation generate image with similar class labels [Che+20]. However, in the context of tumor detection in medical images, cropping could change the semantic meaning if the tumor is cropped [Hua+23].

The astute reader will have noticed that it is trivially possible to perfectly align all pairs  $f(\mathbf{x}), f(\mathbf{x}^+)$  in the representation space by learning a trivial constant function  $f(\mathbf{x})$  (known as a *collapse*). Thus, a core component of joint embedding methods is to define a loss function  $\mathcal{L}$  that prevents  $f$  from learning such degenerate solutions.

Given augmented pairs  $\{(\mathbf{x}_i, \mathbf{x}_i^+)\}_{i=1}^N$ , the *Barlow Twins loss* function [Zbo+21] pushes the cross-correlation matrix  $\mathbf{C} \in \mathbb{R}^{k \times k}$  between the pairs  $f(\mathbf{x}_i)$  and  $f(\mathbf{x}_i^+)$  towards the identity matrix  $\mathbf{I}_k$ . Formally, for a hyperparameter  $\lambda > 0$ , we learn  $f$  by minimizing

$$\mathcal{L}_{BT}(f) = \sum_{j=1}^k (1 - \mathbf{C}_{jj})^2 + \lambda \sum_{j \neq l} \mathbf{C}_{jl}^2 \quad (7)$$

where  $\mathbf{C}$  is the cross-correlation matrix between  $\{f(\mathbf{x}_i)\}_{i=1}^N$  and  $\{f(\mathbf{x}_i^+)\}_{i=1}^N$ , that is

$$\mathbf{C}_{jl} = \frac{\frac{1}{N} \sum_{i=1}^N f_j(\mathbf{x}_i) f_l(\mathbf{x}_i^+)}{\sqrt{\frac{1}{N} \sum_{i=1}^N f_j(\mathbf{x}_i)^2} \sqrt{\frac{1}{N} \sum_{i=1}^N f_l(\mathbf{x}_i^+)^2}}$$

Intuitively, enforcing  $\mathbf{C}_{jj} \approx 1$  ensures that  $f_j(\mathbf{x}_i) \approx f_j(\mathbf{x}_i^+)$  and thus preserves the invariances encoded in the pairs  $\mathbf{x}_i, \mathbf{x}_i^+$ . At the same time, decorrelating  $f_j(\mathbf{x}_i)$  and  $f_j(\mathbf{x}_i^+)$  ensures that each dimension learns a different set of features, preventing dimension collapse.



Since cross-correlation complicates the mathematical treatment of the Barlow Twins loss, theoretical studies use the symmetrized cross-moment matrix [Sim+23; FAG25]

$$\mathbf{C} = \frac{1}{2N} \sum_{i=1}^N \left( f(\mathbf{x}_i) f(\mathbf{x}_i^+)^\top + f(\mathbf{x}_i^+) f(\mathbf{x}_i)^\top \right) \quad (8)$$

to define  $\mathcal{L}_{BT}$  in (7). An improvement over Barlow twins was proposed in [BPL22], called the *Variance-Invariance-Covariance Regularization* or simply the *VICReg loss* function, which appends  $\mathcal{L}_{BT}$  with an invariance term  $\sum_i \|f(\mathbf{x}_i) - f(\mathbf{x}_i^+)\|^2$  to ensure that the embeddings of the augmented pairs are aligned. VICReg has been studied in few theoretical works [Cab+23; FFG24].

Barlow Twins and VICReg models are called examples of *non-contrastive learning* to distinguish them from *contrastive learning*—another class of joint embedding methods, where trivial solutions are avoided by imposing that augmentations from different samples are embedded far apart in the representation space. Two popular contrastive losses are SimCLR [Che+20] and spectral contrastive loss [Hao+21], where the latter is defined as follows.

$$\mathcal{L}_{SCL}(f) = -\frac{2}{N} \sum_{i=1}^n f(\mathbf{x}_i)^\top f(\mathbf{x}_i^+) + \frac{1}{N^2} \sum_{\substack{i,j=1 \\ i \neq j}}^N \left( f(\mathbf{x}_i)^\top f(\mathbf{x}_j^+) \right)^2 \quad (9)$$

The above loss can diverge to  $-\infty$ , which is prevented by projecting the representations onto the unit ball [Hao+21], or by regularizing the norm of the representation [EFG24; FFG24]. There exist numerous empirical studies and heuristic explanations on what these loss functions learn, but they fall short of providing a principled scientific treatment of joint embedding methods, and several open questions remain. In this paper, we address the following questions.

### 1. Characterizing optimal solution

What are the representations or patterns learned by minimizing (7) or (9)?

### 2. Generalization

Do the learned patterns generalize to new data? Can we trust a learned  $f$  to achieve a small  $\mathcal{L}_{BT}$  or  $\mathcal{L}_{SCL}$  loss on new samples?

### 3. Expressivity

Which representations  $f$  can be learned by minimizing  $\mathcal{L}_{BT}$  or  $\mathcal{L}_{SCL}$ ? What is the *ideal* data augmentation that learns  $f$ ?

### 4. Implicit bias of optimization

If we solve the optimization problems in (7) or (9) by gradient descent, then what are the resulting representations?

To answer the first three questions, it is convenient to assume that  $f$  is learned using a kernel model instead of a neural network. Recall that for any positive definite kernel  $\kappa : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ , there exists a corresponding reproducing kernel Hilbert space (rkhs)  $\mathcal{H}$  and a feature map  $\phi : \mathbb{R}^d \rightarrow \mathcal{H}$  such that  $\kappa(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$  [SS02]. A kernel model for representation is a function of the form  $f(\mathbf{x}) = \mathbf{W}\phi(\mathbf{x})$ , where the linear operator  $\mathbf{W} : \mathcal{H} \rightarrow \mathbb{R}^k$  is learned to optimize an objective. When  $\mathcal{H}$  is finite-dimensional, say  $\mathcal{H} = \mathbb{R}^p$ , then  $\mathbf{W} \in \mathbb{R}^{k \times p}$  is simply a matrix with  $k$  rows, each defining one of the  $k$  output dimensions of the function. The kernel setting comes with several advantages. Closed-form expressions of the optimal solution  $f$  can be derived for kernel models, which simplifies questions on expressivity and deriving ideal augmentation. The kernel literature also provides several techniques to estimate the generalization error of the learned  $f$ . Finally, we discuss that if a wide neural network is used to learn the representations, then the implicit bias of gradient descent results in a solution at convergence that is close to a learned kernel model.

### 3.1 Neural tangent kernel regime for joint embedding models

As discussed in Section 1, the *neural tangent kernel (NTK) regime* is well known in the supervised deep learning literature—the learning dynamics of infinitely wide neural networks under gradient descent with small step size is close to that of kernel models [JHG18]. Although this equivalence and error rates for the NTK approximation are known for general loss functions, such results are restricted to a few steps of gradient descent and not until convergence of training [YL21]. Only for specific losses (squared, logistic, the NTK approximation has been studied at convergence [LZB20].

Our interest in the NTK regime stems from the need to characterize the learned representation (discussed later). Hence, we are primarily concerned with the NTK approximation at convergence when minimizing self-supervised loss functions such as (7) or (9). Below, we present the study in [FAG25], which studies the case of Barlow twins loss function (7) with  $\mathbf{C}$  being the symmetrized cross-moment matrix (8) and  $\lambda = 1$ . Observe that the loss function simplifies to  $\mathcal{L}_{BT}(f) = \|\mathbf{C} - \mathbf{I}\|^2$ . We restrict our analysis to the case where  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$  is modeled by a two-layer neural network with hidden-layer width  $M$ ,

$$f(\mathbf{x}) = \frac{1}{\sqrt{M}} \sum_{m=1}^M \mathbf{w}_m \psi(\mathbf{v}_m^\top \mathbf{x}),$$

where  $\mathbf{w}_m \in \mathbb{R}^k$  and  $\mathbf{v}_m \in \mathbb{R}^d$  for all  $m \in [M]$  are trainable parameters, and  $\psi$  is a smooth bounded activation function with bounded first derivative. For example, we could have tanh activation. The weights are initialized as random independent Gaussians with constant variance, and collected in a vector  $\theta \in \mathbb{R}^{M(d+k)}$  that is trained under *gradient flow* (gradient descent with infinitesimally small step sizes)

$$\frac{\partial \theta}{\partial t} = \dot{\theta}(t) = -\frac{\partial \mathcal{L}_{BT}}{\partial \theta}$$

We write  $\theta_0$  for the weights at initialization. The neural tangent kernel (NTK) is defined

as a time-varying, matrix-valued map [JHG18]

$$\mathbf{K}_t(x, x') = \left( \left( \frac{\partial f_i(\mathbf{x})}{\partial \theta(t)} \right)^\top \left( \frac{\partial f_j(\mathbf{x}')}{\partial \theta(t)} \right) \right)_{i,j=1}^k$$

for all  $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ , where  $f_i$  is the  $i$ -th output dimension of  $f$ . To underline the dependence of the NTK on the parameters  $\theta(t)$  that evolve during training, we sometimes also denote it as  $\mathbf{K}_\theta$ . The key insight of the NTK literature, proved mainly for squared loss [JHG18; LZZ20], is that the NTK does not change during training if the width of the neural network approaches infinity. Consequently, the training dynamics of  $f$  approach those of kernel regression with respect to the (vector-valued) kernel at initialization  $\mathbf{K}_0$ . The constancy of the NTK in the infinite width limit essentially relies on three facts:

1. The spectral norm of the Hessian of the neural network is  $\mathcal{O}\left(\frac{R}{\sqrt{M}}\right)$  for all weights  $\theta$  with  $\|\theta - \theta_0\| \leq R$ .
2. The change in the NTK from  $\theta_0$  to any  $\theta$  can be bounded in terms of the Hessian, and  $\|\theta - \theta_0\|$ .
3.  $R$  is independent of  $M$  because convergence happens in a ball of width-independent radius around  $\theta_0$ .

The first fact is true regardless of the loss function. The same is true for the second fact. However, the third piece in the puzzle is missing: Unless  $R$  remains independent of  $M$ , we do not obtain constancy of the NTK at a large width  $M \rightarrow \infty$ . To ensure that this holds, it is instructive to look at the evolution of the loss  $\mathcal{L}_{BT}$  and the parameters *over time*. Defining

$$\mathbf{u}(t) = \begin{bmatrix} \left( \frac{\partial \mathcal{L}_{BT}}{\partial f_j(\mathbf{x}_i)} \right)_{i,j} \\ \left( \frac{\partial \mathcal{L}_{BT}}{\partial f_j(\mathbf{x}_i^+)} \right)_{i,j} \end{bmatrix}, \quad \mathbf{K}(t) = \begin{bmatrix} \mathbf{K}_t(x_1, x_1) & \dots & \mathbf{K}_t(x_1, x_N^+) \\ \vdots & \ddots & \vdots \\ \mathbf{K}_t(x_N^+, x_1) & \dots & \mathbf{K}_t(x_N^+, x_N^+) \end{bmatrix}$$

the time evolution of the Barlow Twins loss can be expressed as

$$\frac{\partial}{\partial t} \mathcal{L}_{BT}(t) = -\mathbf{u}(t)^\top \mathbf{K}(t) \mathbf{u}(t).$$

Under certain assumptions, namely that the loss at initialization is smaller than 1 and that the smallest eigenvalue of the kernel matrix at initialization is safely bounded away from zero, it can be shown that  $\mathcal{L}_{BT}(t)$  decreases exponentially fast for **all** networks of sufficiently large width. Consequently, we may pick a time  $T$  that is width-independent and ensures convergence of training until  $T$ . Additionally, it can be shown that for any  $\epsilon > 0$ , there exists a width-independent  $\kappa > 0$  such that  $\sup_{t \leq T} \|\dot{\theta}(t)\| \leq \kappa$  holds with high probability  $\geq 1 - \epsilon$  for any network of sufficiently large width, implying that the weights cannot have moved too much until convergence. Together, both of these observations yield the following result.

**Theorem 3.1** (Constancy of the NTK under Barlow Twins loss minimization [FAG25]). *Assume that for any network of sufficiently large width, the NTK matrix  $\mathbf{K}(0)$  is positive definite with the smallest eigenvalue  $\lambda_{\min}(\mathbf{K}(0)) \geq \lambda > 0$ , and that  $\mathcal{L}_{BT}(0) \leq 1 - \rho < 1$  for some small  $\rho > 0$ . Then, there exists a real number  $R > 0$  such that, with probability at least  $1 - \epsilon$ , the change of the NTK until convergence of the loss (up to small  $\delta > 0$ ) is  $\mathcal{O}(R^2/\sqrt{M})$ . In particular, the radius  $R$  depends only on  $\delta, N, \lambda, \rho, \epsilon$ , but not on the network width.*

Figure 5 illustrates our theoretical results empirically: As the width of the network increases, the NTK changes less and less, and consequently the representations learned in a neural network approach those of a kernel model.

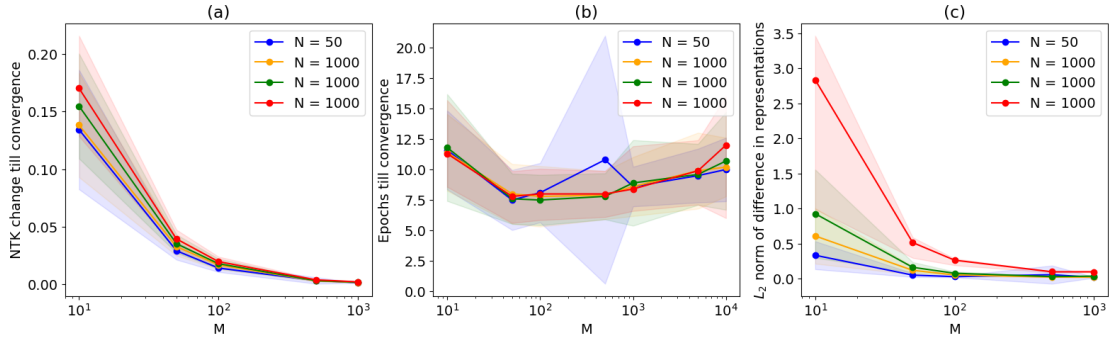


Figure 5: Source [FAG25]. **Numerical evidence of constancy of NTK under Barlow Twins loss minimization.** We verify Theorem 3.1 by training a 1-hidden layer network with tanh activation on the MNIST dataset. We use gradient descent with a learning rate of 0.5, and train till loss  $\mathcal{L}_{BT} \leq 10^{-5}$ . The results are averaged over 10 independent runs. For a fixed sample size  $N$ , we plot different quantities for varying network width  $M$ . We then vary  $N$  and plot: (a) NTK change till convergence, where we see that as width increases, there is less change in NTK between initialization and convergence; (b) training epochs till convergence, which shows that the time to convergence remains almost constant with the network width; (c) squared norm of difference between representations of neural network and corresponding kernel model under the Barlow Twins loss, which validates that one can use the optimal solution of the kernel model (see Section 3.2) as a good approximation for the representation learned by a neural network.

In this section, we consider only the training dynamics of networks under Barlow twins loss. It is natural to ask if a similar conclusion can be drawn for other self-supervised loss functions, for example VICReg or spectral contrastive loss. In [FAG25], we discuss how the analysis could be extended to other loss functions. A similar analysis for contrastive loss functions is presented in [AEG25], where it is also shown that the NTK equivalence does not hold for all loss functions. For a simple contrastive loss function, the output of the neural network diverges to infinity within a logarithmic number of gradient descent steps. Our results on the convergence or divergence of training dynamics in [FAG25; AEG25] are particularly interesting because prior work implicitly assumes that the NTK equivalence always holds for self-supervised models. In particular, [Sim+23] use the kernel equivalence to show that representations are learned in a stepwise manner, one dimension at a time, while [Cab+23] used the kernel connection to study generalization properties and inductive biases of VICReg.

There also exist a few works beyond the NTK regime, focusing on specific settings. [WL21;

[WL22] study the learning dynamics of the joint embedding model with a 1-hidden layer ReLU network under sparse coding and strong/weak feature models and characterize the features learned. [Tia22] study the dynamics of contrastive learning under coordinate-wise optimization.

### 3.2 Kernel-based self-supervised models learn spectral projections

In this section, we focus on characterizing the learned representations  $\mathbf{x} \mapsto f(\mathbf{x})$ , assuming that  $f$  is a kernel model. Note that the NTK equivalence of trained neural networks at convergence, discussed in the previous section, allows us to interpret the subsequent discussion in the context of deep learning—we do not explicitly consider neural networks in this section. Several works have characterized the optimal solution for various self-supervised loss functions, often assuming linear or kernel models. The essence of such a characterization could be summarized as “*optimal representations learned by self-supervised models are spectral projections*”. However, the results differ somewhat in terms of the matrix or operator whose spectral decomposition is used. The most notable work is [BL22], which provides a unified framework that relates several self-supervised loss functions and argues that any of the loss minimization is related to spectral embedding. [Sim+23; Cab+23] provide more precise characterization of Barlow Twins and VICReg loss minimization, respectively, in terms of spectral decomposition of a cross-covariance operator. In [EFG24], we provide a corresponding result for several contrastive loss functions, including the spectral contrastive loss. Using a slightly different perspective, [Hao+21; JHM23] show that joint embedding methods learn spectral projections of a *augmentation graph*—a graph over all unlabeled augmented data with edges connecting the augmented pairs.

In line with the earlier discussions on the Barlow Twins loss, we state a characterization of the representations learned by minimizing  $\mathcal{L}_{BT}$ , adapted from [Sim+23]. Recall that the loss is given by  $\mathcal{L}_{BT}(f) = \|\mathbf{C} - \mathbf{I}_k\|^2$ , where  $\mathbf{C} \in \mathbb{R}^{k \times k}$  is the empirical cross-moment matrix of  $f(\mathbf{x}), f(\mathbf{x}^+)$ . But under a kernel model  $f(\mathbf{x}) = \mathbf{W}\phi(\mathbf{x})$ , it can be rephrased as

$$\mathcal{L}_{BT}(\mathbf{W}) = \|\mathbf{W}\Gamma\mathbf{W}^\top - \mathbf{I}_k\|^2, \quad \text{where } \Gamma = \frac{1}{2N} \sum_{i=1}^N \phi(\mathbf{x}_i)\phi(\mathbf{x}_i^+)^\top + \phi(\mathbf{x}_i^+)\phi(\mathbf{x}_i)^\top$$

is the symmetrized cross-moment matrix of the feature map  $\phi : \mathbb{R}^d \rightarrow \mathcal{H}$ . Note that we use the notation  $\mathbf{a}\mathbf{b}^\top$  in the sense of an outer product. The following result holds.

**Theorem 3.2** (Optimal representations learned by kernel Barlow Twins model, adapted from [Sim+23]). *Let the eigen pairs of  $\Gamma$  be given by  $(\lambda_i, \mathbf{u}_i)_{i=1,2,\dots}$  in decreasing order of*

*eigenvalues. If  $\lambda_k > 0$ , the projection  $\mathbf{W}^* : \mathcal{H} \rightarrow \mathbb{R}^k$  given by  $\mathbf{W}^* = \mathbf{Q} \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} \mathbf{u}_1^\top \\ \vdots \\ \frac{1}{\sqrt{\lambda_k}} \mathbf{u}_k^\top \end{bmatrix}$ , where*

*$\mathbf{Q} \in \mathbb{R}^{k \times k}$  is any orthogonal matrix, achieves  $\mathcal{L}_{BT} = 0$ . In particular,  $\mathbf{W}^*$  is the minimum norm solution that achieves zero loss, and under some conditions on initialization, the gradient descent converges to the solution  $\mathbf{W}^*$ .*

The learned representation can be expressed in terms of the kernel function  $\kappa$  in the following way. Let  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  and  $\mathcal{X}^+ = \{\mathbf{x}_1^+, \dots, \mathbf{x}_N^+\}$  refer to the unlabeled data and their respective augmentation, and  $\mathbf{x}$  be the new sample for which we compute  $f(\mathbf{x}) = \mathbf{W}^* \phi(\mathbf{x})$ . We use  $\mathbf{K}_{\mathcal{X}\mathcal{X}^+} \in \mathbb{R}^{N \times N}$  denote a matrix with  $[\mathbf{K}_{\mathcal{X}\mathcal{X}^+}]_i = \kappa(\mathbf{x}_i, \mathbf{x})$  and analogously define  $\mathbf{K}_{\mathcal{X}\mathcal{X}}, \mathbf{K}_{\mathcal{X}^+\mathcal{X}^+}, \mathbf{K}_{\mathcal{X}^+\mathcal{X}} \in \mathbb{R}^{N \times N}$  and  $\mathbf{K}_{\mathcal{X}\mathbf{x}}, \mathbf{K}_{\mathcal{X}^+\mathbf{x}} \in \mathbb{R}^{N \times 1}$ . Define the matrices

$$\mathbf{Z} = \frac{1}{2N} \left( \begin{bmatrix} \mathbf{K}_{\mathcal{X}\mathcal{X}^+} \\ \mathbf{K}_{\mathcal{X}^+\mathcal{X}^+} \end{bmatrix} \begin{bmatrix} \mathbf{K}_{\mathcal{X}\mathcal{X}} & \mathbf{K}_{\mathcal{X}\mathcal{X}^+} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{\mathcal{X}\mathcal{X}} \\ \mathbf{K}_{\mathcal{X}^+\mathcal{X}} \end{bmatrix} \begin{bmatrix} \mathbf{K}_{\mathcal{X}^+\mathcal{X}} & \mathbf{K}_{\mathcal{X}^+\mathcal{X}^+} \end{bmatrix} \right),$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathcal{X}\mathcal{X}} & \mathbf{K}_{\mathcal{X}\mathcal{X}^+} \\ \mathbf{K}_{\mathcal{X}^+\mathcal{X}} & \mathbf{K}_{\mathcal{X}^+\mathcal{X}^+} \end{bmatrix} \quad \text{and} \quad \mathbf{K}_\Gamma = \mathbf{K}^{-1/2} \mathbf{Z} \mathbf{K}^{-1/2}.$$

The learned representations are given by

$$f(\mathbf{x}) = \mathbf{W}^* \phi(\mathbf{x}) = \mathbf{Q} \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} \mathbf{v}_1^\top \\ \vdots \\ \frac{1}{\sqrt{\lambda_k}} \mathbf{v}_k^\top \end{bmatrix} \mathbf{K}^{-1/2} \begin{bmatrix} \mathbf{K}_{\mathcal{X}\mathbf{x}} \\ \mathbf{K}_{\mathcal{X}^+\mathbf{x}} \end{bmatrix},$$

with  $\mathbf{Q} \in \mathbb{R}^{k \times k}$  any orthogonal matrix and  $(\lambda_i, \mathbf{v}_i)$  eigen pairs of  $\mathbf{K}_\Gamma$  (which has same nonzero eigenvalues as  $\Gamma$ ).

The first part of the theorem asserts that diagonalizing the cross-moment matrix  $\Gamma$  is done best (in a least norm sense) by projecting onto the top eigenvectors, which is also the solution learned by gradient descent due to *implicit bias*. Note that if  $\lambda_k \leq 0$ , it is impossible to achieve zero loss, since  $\mathbf{W} \Gamma \mathbf{W}^\top$  cannot have more positive eigenvalues than  $\Gamma$  itself. The second part of the result provides the practically implementable representation in terms of the kernel function  $\kappa$  using the so-called *kernel trick*. Note that the resulting solution is bit more complicated in comparison to kernel ridge regression or kernel PCA. This is because the Barlow Twins loss  $\mathcal{L}_{BT}$  is *quartic* (degree 4) in the parameters  $\mathbf{W}$  instead of quadratic objectives of PCA or squared regression.

To summarize the above discussion, we see that in a kernel setting, it becomes possible to precisely characterize the (minimum norm) solutions of Barlow Twins, which interestingly reveals that  $\mathcal{L}_{BT}$  essentially projects onto the dominant eigen space of the cross-moment matrix between positive pairs  $(\mathbf{x}, \mathbf{x}^+)$ , thereby generalizing classical unsupervised representation learning methods like kernel PCA. This underlines the vital role of the augmentation in determining the patterns learned in joint embedding methods, which we investigate next.

### 3.3 Optimal data augmentation to express desired representation

The choice of augmentations is crucial to the performance of self-supervised representation learning, since different downstream tasks may require dramatically different augmentations. The literature on visual foundation models contains several empirical works that study which augmentations work better and how many different augmentations are needed (see the discussions in [FPG24]). For example, it is known that while cropping encourages

invariance to occlusions, it also negatively affects downstream tasks that require category and viewpoint invariance [PG20]. There is limited theoretical understanding regarding the subtleties of the augmentation choice.

In particular, a fundamental question remains unanswered: *What is the relationship between data, loss function, and the augmentations needed to learn a desired target representation  $f^*$ ?* Observe that the question is similar to the notion of *expressivity* or *universal approximation* [DHP21]. However, unlike supervised learning, where the model has access to the target representation through labeled data  $\{(\mathbf{x}_i, f^*(\mathbf{x}_i))\}_{i=1}^N$ , the above question asks if one can learn  $f^*$  only with access to unlabeled augmented pairs  $\{(\mathbf{x}_i, \mathbf{x}_i^+)\}_{i=1}^N$ .

We address the above question in [FFG24], which also makes practical considerations by allowing random data augmentation. For ease of exposition, we consider a simpler setup of deterministic augmentation and adapt the results of [FFG24] accordingly.

**Formal problem on expressivity of self-supervised models.** Given unlabeled data  $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^d$  and a target representation  $f^* : \mathbb{R}^d \rightarrow \mathbb{R}^k$ , does there exist an augmentation map  $\mathbf{x}_i \rightarrow \mathbf{x}^+ = T(\mathbf{x})$  such that the representation  $f$  learned by minimizing  $\mathcal{L}_{BT}$  (7) or  $\mathcal{L}_{SCL}$  (9) using the augmented pairs  $\{(\mathbf{x}_i, T(\mathbf{x}_i))\}_{i=1}^N$  is identical to  $f^*$ , up to affine transformation?

Note that it suffices to learn  $f^*$  up to the affine transformation since, in practice, a linear predictor is typically fitted using the learned embedding  $f^*(\mathbf{x})$  for downstream prediction tasks (see, for example, Figure 1). Assuming that  $f$  is learned using a kernel self-supervised model, we can exploit the characterization of the optimal representation (such as Theorem 3.2) to derive the optimal data augmentation. To this end, we relax the above-mentioned problem and try to prove the existence of a transformation  $T_{\mathcal{H}} : \mathcal{H} \rightarrow \mathcal{H}$ ,  $\phi(\mathbf{x}_i) \mapsto \phi_i^+ \in \mathcal{H}$  instead of finding the augmentation in the input space  $\mathbf{x}_i \mapsto \mathbf{x}_i^+$ , where  $\mathcal{H}$  in the reproducing kernel Hilbert space (rkhs)  $\mathcal{H}$  associated with the kernel  $\kappa$ . [FFG24] characterizes  $T_{\mathcal{H}}$  for  $\mathcal{L}_{BT}$  and  $\mathcal{L}_{SCL}$ , which we state below.

**Theorem 3.3** (Optimal data augmentation for spectral contrastive and Barlow Twins models [FFG24]). *Let  $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^d$  be given unlabeled data,  $f^* : \mathbb{R}^d \rightarrow \mathbb{R}^k$  be a target representation, and  $\kappa : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$  be a pre-specified kernel with associated rkhs  $\mathcal{H}$  and map  $\phi : \mathbb{R}^d \rightarrow \mathcal{H}$ . Define  $\Phi = [\phi(\mathbf{x}_1) \ \dots \ \phi(\mathbf{x}_N)]$ .*

*Assume that  $f^* = \mathbf{S}\Phi^\top$  and that both the kernel matrix  $\mathbf{K} = \Phi^\top\Phi \in \mathbb{R}^{N \times N}$  and the matrix  $\mathbf{S}\mathbf{K}\mathbf{S}^\top \in \mathbb{R}^{k \times k}$  have full rank. The following statements hold.*

- **Spectral contrastive:** *Define the augmentation  $T_{\mathcal{H}} : \mathcal{H} \rightarrow \mathcal{H}$  such that for  $\mathbf{x} \in \mathbb{R}^d$ ,*

$$\phi^+ = T_{\mathcal{H}}(\phi(\mathbf{x})) = \Phi \mathbf{S}^\top (\mathbf{S} \mathbf{K} \mathbf{S}^\top)^{-1} \mathbf{S} \Phi^\top \phi(\mathbf{x}).$$

*If  $f$  is the least norm minimizer of  $\mathcal{L}_{SCL}$  (9) with additional regularization  $\frac{1}{N} \sum_{i=1}^N \|f(\mathbf{x}_i)\|^2 + \|f(\mathbf{x}_i^+)\|^2$  and augmented pairs  $\{(\phi(\mathbf{x}_i), \phi_i^+)\}_{i=1}^N$ , then  $f = f^*$  up to affine transformation.*



- **Barlow Twins:** Define the augmentation  $T_{\mathcal{H}} : \mathcal{H} \rightarrow \mathcal{H}$  such that for any  $\mathbf{x} \in \mathbb{R}^d$ ,

$$\phi^+ = T_{\mathcal{H}}(\phi(\mathbf{x})) = \Phi \mathbf{K}^{-1/2} \mathbf{B} \mathbf{K}^{-1/2} \Phi^\top \phi(\mathbf{x}),$$

where  $\mathbf{B}$  is a solution to the Lyapunov equation

$$\mathbf{K} \mathbf{B} + \mathbf{B} \mathbf{K}^\top = 2N \cdot \mathbf{K}^{1/2} \mathbf{S}^\top \left( \mathbf{S} \mathbf{K} \mathbf{S}^\top \right)^{-2} \mathbf{S} \mathbf{K}^{1/2}$$

If  $f$  is the least norm minimizer of  $\mathcal{L}_{BT}$  (7) with augmented pairs  $\{(\phi(\mathbf{x}_i), \phi_i^+)\}_{i=1}^N$ , then  $f = f^*$  up to affine transformation.

It is important to discuss the assumptions made in the above theorem. The assumption that  $\mathbf{K}$  is full rank holds for any universal kernel (for example, Gaussian, Laplace, etc.) and distinct points  $\mathbf{x}_1, \dots, \mathbf{x}_N$ . Assuming that the target  $f^* = \mathbf{S} \Phi^\top$ , or rather that  $f^* \in \text{span}\{\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)\}$  ensures that  $f^*$  is itself a minimum norm solution since, by virtue of the representer theorem [SS02], any least norm minimizer of a loss function is contained in the span. One could extend the theorem to the cases  $f^* \notin \text{span}\{\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)\}$  or  $f^* \notin \mathcal{H}$  by accounting for the error in projecting on to the span, which can be made arbitrarily small by using universal kernels and allowing arbitrarily large  $N$ . Finally, the full rank of  $\mathbf{S} \mathbf{K} \mathbf{S}^\top$  implies that the sample covariance of  $\{f^*(\mathbf{x}_i)\}_{i=1}^N$  is full rank.

To summarize the above discussion, we show that under certain conditions, a target  $f^*$  can be learned through self-supervised learning using specific augmentations in the rkhs. It is easy to see that, in the context of spectral contrastive models, the augmentation  $T_{\mathcal{H}}$  is simply a rank- $k$  projection in  $\mathcal{H}$ . It is natural to ask if one can derive the corresponding augmentation  $\mathbf{x} \mapsto \mathbf{x}^+$  in the input space. This may not be possible, in general, as the map  $\phi : \mathbb{R}^d \rightarrow \mathcal{H}$  may not be surjective. However, [FFG24] presents an approach to numerically find approximate augmentations in the input space by solving a *kernel pre-image problem*. Figure 6 illustrates the optimal augmentation for different kernel models, which are surprisingly quite different from commonly used augmentations like rotation or cropping.

### 3.4 Statistical generalization for learned representations

We return to the topic of statistical generalization, which is the central theme of this paper. The limiting kernel regime for self-supervised neural networks as well as the characterization of the learned representations in the kernel setting help to derive generalization error bounds for joint embedding models. Note that the question of generalization needs to be studied in two separate contexts. In this section, we study the generalization of the learned representation to new samples with respect to a self-supervised loss, (7) or (9). In the next section, we discuss the pertinent topic of the generalization error of downstream predictors.

In supervised settings or the case of reconstruction, it makes sense to bound the generalization error defined as the expected loss on a new sample. However, this notion of generalization may not be meaningful in the context of self-supervised losses. For example, consider the case of simplified Barlow Twins loss  $\mathcal{L}_{BT}(f) = \|\mathbf{C} - \mathbf{I}_k\|^2$  and assume that there is

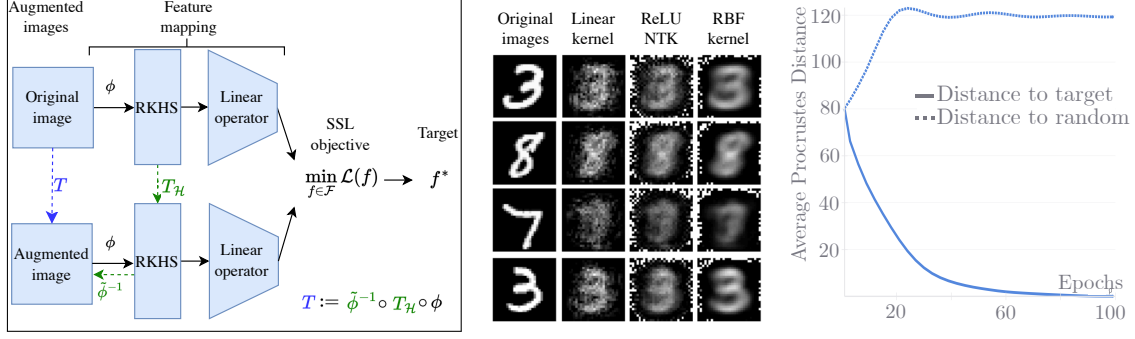


Figure 6: Source [FFG24]. **Optimal data augmentation for kernel spectral contrastive model.** The left plot shows a schematic for constructing augmentation  $\mathbf{x} \mapsto \mathbf{x}^+$  by using the optimal augmentation  $T_{\mathcal{H}}$  in the rkhs  $\mathcal{H}$ . The middle plot compares augmented MNIST images using different kernels, demonstrating different function classes require different augmentations to achieve the same representations. Here, the target representation  $f^*$  is considered as the one obtained from a pre-trained ResNet50. The right plot shows that average Procrustes distance between the learned representation  $f$  from target  $f^*$  and random representations during training given the augmentations  $\mathcal{T}_{\mathcal{H}}$ . It validates our theory that the analytically derives  $T_{\mathcal{H}}$  indeed results in learning the target representation  $f^*$ .

an underlying distribution for the augmented pairs  $(\mathbf{x}, \mathbf{x}^+)$ . The natural notion of generalization error in this case would be  $\mathbb{E}_{(\mathbf{x}, \mathbf{x}^+)} \left[ \left\| \frac{1}{2} (f(\mathbf{x})f(\mathbf{x}^+)^{\top} + f(\mathbf{x}^+)f(\mathbf{x})^{\top}) - \mathbf{I}_k \right\|^2 \right]$ , which is always at least  $k - 2$  since the cross-covariance term has rank 2. In [FAG25], we avoid this issue by deriving error bounds for a different notion of a population loss for Barlow Twins.

It is possible to state generalization error bounds for the spectral contrastive loss (9) using classical uniform convergence results, as we state below.

**Theorem 3.4** (Generalization error bound for kernel spectral contrastive model, adapted from [EFG24]). *Assume that the kernel  $\kappa$  is bounded with  $\nu := \sup_{\mathbf{x}} \kappa(\mathbf{x}, \mathbf{x})$  and the augmented pairs  $\{(\mathbf{x}_i, \mathbf{x}_i^+)\}_{i=1}^N$  are independent and identically distributed samples from a distribution  $\mathcal{D}$  on  $\mathbb{R}^d \times \mathbb{R}^d$ .*

*Let  $\hat{f} = \arg \min_{f \in \mathcal{F}} \mathcal{L}_{SCL}(f)$ , where  $\mathcal{L}_{SCL}(f)$  is the spectral contrastive loss in (9) and the optimization is over the hypothesis class of kernel models with a bounded norm  $\mathcal{F} = \{f(\mathbf{x}) = \mathbf{W}\phi(\mathbf{x}) : \|\mathbf{W}\| \leq \omega\}$ . Let  $\mathcal{L}'_{SCL}$  denote the loss for an independent set of pairs  $\{(z_i, z_i^+)\}_{i=1}^n \sim \mathcal{D}^n$ . With probability  $1 - \delta$*

$$\mathbb{E}_{\{(z_i, z_i^+)\}_{i=1}^n} [\mathcal{L}'_{SCL}(\hat{f})] \leq \mathcal{L}_{SCL}(\hat{f}) + O \left( \omega^3 \kappa^2 \sqrt{\frac{k + \omega^2 \log \frac{1}{\delta}}{N}} \right).$$

## 4 Statistical performance guarantees for downstream predictors

In the modern context of foundation models, unsupervised representation learning as a *pretraining* phase. Good representations help reduce the number of labeled samples required to train supervised predictors. A well-known example of this is *principal component*

*regression*, where one first uses PCA to reduce the dimension of training data and then fits the regressor. PCA has been established to act as a regularizer for regression, specifically when the true signal is aligned with the principal components [DFH17]. Hence, statistically, alternative (self-supervised) approaches are needed when the ideal representations for downstream prediction do not align with the principal components of the unlabeled data. Theorem 3.3 establishes that, with appropriate augmentation, we can learn any target representation. Hence, the general strategy for deriving statistical guarantees for downstream generalization is to integrate the aforementioned results into a statistical framework.

In recent years, two distinct principles have been used to derive downstream generalization error bounds. [Sau+19] introduced a *contrastive unsupervised representation learning (CURL)* framework, which can be applied to general joint embedding methods. Here, one assumes that the augmented unlabeled data is sampled from a mixture distribution with augmented pairs having a higher probability of being sampled from the same class. [Hao+21] propose an alternative framework that presents generalization error bounds in terms of properties of a population augmentation graph, where edges connect samples from same class. Despite being more general than CURL, the augmentation graph-based framework is limited, since the population and the empirical augmentation graphs are not known to be close. We are also unaware of tight error bounds that arise from this framework. Hence, in this section, we present results based on the CURL framework [Sau+19; vG24].

The statistical framework in CURL [Sau+19] is formalized as follows. Assume that there are  $p$  classes with respective class conditional distributions  $\mathcal{D}(\cdot|y)$  on  $\mathbb{R}^d$ , where  $y \in [p]$ . It is assumed that each augmented pair  $(\mathbf{x}, \mathbf{x}^+)$  consists of samples of the same class. Using the above setup, one can derive uniform convergence bounds [Sau+19; EFG24] or tighter PAC-Bayesian bounds [vG24] on the self-supervised loss—the latter bounds also avoid the independence assumption about samples in augmented pair. Once a representation  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$  is learned by pre-training, a linear classifier is assumed to be trained for label prediction. The idea is to relate the downstream prediction error to the self-supervised loss, where the tightness of the bounds depends on the losses considered. We present a result from [vG24] that relates the SimCLR contrastive loss [Che+20] to the supervised cross-entropy loss. We first define the two loss functions. For augmented data  $\{(\mathbf{x}_i, \mathbf{x}_i^+)\}_{i=1}^N$ , the SimCLR loss is given by

$$\mathcal{L}_{\text{SimCLR}}(f) = \frac{1}{N} \sum_{i=1}^N -\log \frac{\exp\left(\frac{f(\mathbf{x}_i)^\top f(\mathbf{x}_i^+)}{\tau}\right)}{\exp\left(\frac{f(\mathbf{x}_i)^\top f(\mathbf{x}_i^+)}{\tau}\right) + \sum_{j \neq i} \exp\left(\frac{f(\mathbf{x}_i)^\top f(\mathbf{x}_j^+)}{\tau}\right)}, \quad (10)$$

where  $\tau > 0$  is a temperature scaling hyperparameter that significantly impacts performance (see Table 1). On the other hand, for a labeled sample  $(\mathbf{x}, y) \in \mathbb{R}^d \times [p]$  and a learned embedding  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$ , assume that a  $p$  class classifier is characterized by the matrix  $\mathbf{A} = [a_1, \dots, a_p] \in \mathbb{R}^{k \times p}$ , where the prediction for class- $j$  uses the projection

$f(\mathbf{x})^\top a_j$ . The cross entropy loss is given by

$$\mathcal{L}_{CE}(f, \mathbf{A}) = -\log \frac{\exp(f(\mathbf{x})^\top a_y)}{\sum_{y'=1}^p \exp(f(\mathbf{x})^\top a_{y'})}. \quad (11)$$

**Theorem 4.1** (Generalization error bounds for supervised cross entropy loss in terms of SimCLR loss [vG24]). *Suppose the labeled data distribution  $\mathcal{D}$  on  $\mathbb{R}^d \times [p]$  is characterized by  $\pi_j := \mathbb{P}_{(\mathbf{x}, y) \sim \mathcal{D}}(y = j)$ . Define  $\Delta = \log\left(\frac{p}{N-1} \cosh^2\left(\frac{1}{\tau}\right)\right)$  and  $\alpha = \log p + \min\{0, \log(\cosh^2(1)) - \tau\Delta\}$ . Then for any representation  $f : \mathbb{R}^d \rightarrow \mathbb{S}^{k-1}$  (unit sphere in  $\mathbb{R}^k$ ),*

$$\begin{aligned} & \min_{\mathbf{A} \in \mathbb{R}^{k \times p}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [\mathcal{L}_{CE}(f, \mathbf{A})] \\ & \leq \min \left\{ \frac{\sigma}{\tau} + \Delta + \mathbb{E}[\mathcal{L}_{SimCLR}(f)], \sigma + \tau\Delta + \alpha + \tau \mathbb{E}[\mathcal{L}_{SimCLR}(f)] \right\}. \end{aligned}$$

Here,  $\sigma = \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} \|\mathbf{f}(\mathbf{x}) - a_y^*\| \leq 2$ , where  $a_y^*$  denotes the population mean of the representation  $f$ , conditioned on label  $y$ . The expectation of  $\mathcal{L}_{SimCLR}$  is with respect to the unlabeled samples used in pre-training.

The above result shows that the cross-entropy generalization error of the optimal linear classifier is bounded by the population SimCLR loss along with few additional terms. It is natural to question the utility of Theorem 4.1. In practice, the bound can be incorporated into a PAC-Bayesian framework to derive risk certificates (see the numerical results in Table 1). As shown in [BNN22], such bounds lead to non-vacuous error bounds that reflect the general trends of the test error much better than uniform convergence bounds (see Figure 1 in [BNN22]). Theorem 4.1 is a refinement of [BNN22] and is tighter for a low temperature  $\tau$ .

		SimCLR loss			Cross entropy (supervised)			
		$\tau = 1.0$	$\tau = 0.5$	$\tau = 0.2$	Test loss	$\tau = 1.0$	$\tau = 0.5$	$\tau = 0.2$
Test loss		4.9	4.26	2.71	1.76	1.70	1.70	
Bounds	PAC-Bayes i.i.d.	8.48	8.91	10.17	[BNN22]	<b>3.27</b>	5.30	12.59
	$f$ -divergence [NGG20]	27.03	33.27	48.47	[vG24, Theorem 5]	<b>3.27</b>	<b>4.95</b>	<b>4.61</b>
	[vG24, Theorem 3]	<b>5.20</b>	6.27	43.77				
	[vG24, Theorem 4]	5.54	<b>5.49</b>	<b>6.22</b>				

Table 1: Source [vG24]. **Tight PAC-Bayesian generalization error bounds for the SimCLR loss and downstream supervised cross-entropy loss.** The empirical results and the bounds are for a 7-layer convolutional network pretrained on CIFAR-10 data set. The columns on the left show the empirical SimCLR loss and respective theoretical bounds using classical PAC-Bayesian bounds with the incorrect assumption that all samples are independent,  $f$ -divergence based PAC-Bayes bounds from [NGG20] and our bounds [vG24] using a McAllester-type and KL-divergence-based PAC-Bayes bounds. The latter three bounds account for sample dependence. The temperature scaling  $\tau$  heavily influences the empirical SimCLR loss and only one of bounds are non-vacuous at small temperature. The columns on the right show the downstream test error and generalization error bounds for the cross-entropy loss. Our bounds [vG24] improve up existing bounds at small temperature.

## 5 Future directions and open questions

The statistical theory of unsupervised representation learning is still in a nascent stage. Compared with supervised deep learning theory, this direction of research is more complex and interesting because of the inherent challenges of formalizing unsupervised learning. However, there is significantly less attention to this topic from the machine learning theory community. The most surprising fact is that the most recent theoretical advances related to statistical generalization in foundation models, such as [YH21; Bah+24], mainly focus on supervised settings and do not address open questions about unsupervised representation learning or generalization guarantees for downstream prediction. This paper attempts to provide an overview of recent theoretical research on unsupervised (deep) representation learning. Although many of the aforementioned results are taken from our recent work, there are also other interesting works on this topic, some of which are cited above. We conclude this paper with some important open questions about this topic, beginning with the crucial question of precise error rates.

**Optimal generalization error rates.** One of the main goals in statistics is to characterize minimum possible error rates and derive the optimal methods. Although Theorems 3.4–4.1 provide upper bounds on self-supervised or downstream generalization errors, they do not provide a precise characterization like Theorems 2.2–2.3. In particular, lower bounds of the generalization error are needed in the context of foundation models. Recent work [Den+24; Ge+24] has taken key steps in studying self-supervised learning from the perspective of meta learning and maximum likelihood estimation, which provide a statistical framework to study lower bounds of error and lead to the design of optimal approaches. We also believe that the kernel self-supervised model described in this paper can help to derive precise error rates similar to the results on principal component regression [DFH17].

The other notable open direction is to understand the role of neural networks in unsupervised representation learning. We may formally pose the following questions.

**Role of feature and attention learning in unsupervised models.** Recent literature on supervised learning has studied the statistical advantage of using deep neural networks or transformer architectures. For example, in the context of learning a  $k$ -parity function, [DM20; HG25] proves that, compared to kernel models, trained 1-hidden-layer networks or transformers require significantly fewer parameters. In a similar vein, one could ask whether training a non-linear autoencoder or joint embedding model provably recovers the signal better than kernel models.

A more precise understanding of unsupervised representation learning and its impact on downstream predictors will not only help in the theory of foundation models but also significantly advance the fields of statistics and theoretical machine learning as a whole.

**Acknowledgment and funding information.** The research contribution of P. M. Esser, mentioned in this paper, was carried out when he was at the Technical University of Munich, supported by the German Research Foundation (DFG) Priority Program 2298 “Theoretical Foundations of Deep Learning” (project GH-257/2-1). M. Fleissner is supported by the DAAD program Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Education and Research (BMBF). In addition, D. Ghoshdastidar acknowledges the German Research Foundation for funding through the project GH-257/4-1, DFG-ANR PRCI “ASCAI” (project GH-257/3-1) and DFG GRK 2428 “ConVeY”. The authors thank Gilles Blanchard, Leena Chennuru Vankadara, Satyaki Mukherjee, Mahalakshmi Sabanayagam, Maedeh Zarvandi, Gautham Govind Anil, Shlomo Libo Feigin, Jonghyun Ham, Anna van Elst and Theresa Wasserer, who have collaborated on some of the cited works or are currently collaborating on this topic.

## References

- [AEG25] Gautham Govind Anil, Pascal Mattia Esser, and Debarghya Ghoshdastidar. “When can we Approximate Wide Contrastive Models with Neural Tangent Kernels and Principal Component Analysis?” In: *Thirty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2025, Philadelphia, USA*. Ed. by Toby Walsh, Julie Shah, and Zico Kolter. AAAI Press, 2025.
- [AP21] Sercan Ö. Arik and Tomas Pfister. “TabNet: Attentive Interpretable Tabular Learning”. In: *AAAI Conference on Artificial Intelligence*. 2021.
- [Aro+19] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. “Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks”. In: *International Conference on Machine Learning*. 2019.
- [Ba+22] Jimmy Ba, Murat A. Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. “High-dimensional Asymptotics of Feature Learning: How One Gradient Step Improves the Representation”. In: *Advances in Neural Information Processing Systems, NeurIPS*. 2022.
- [Bah+24] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. “Explaining neural scaling laws”. In: *Proceedings of the National Academy of Sciences* (2024).
- [Bao+20] Xuchan Bao, James Lucas, Sushant Sachdeva, and Roger B Grosse. “Regularized linear autoencoders recover the principal components, eventually”. In: *Advances in Neural Information Processing Systems*. 2020.
- [BAP24] Blake Bordelon, Alexander B. Atanasov, and Cengiz Pehlevan. “A Dynamical Model of Neural Scaling Laws”. In: *International Conference on Machine Learning, ICML*. 2024.
- [BCM05] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. “A review of image denoising algorithms, with a new one”. In: *Multiscale modeling & simulation* (2005).

- [BCP20] Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. “Spectrum Dependent Learning Curves in Kernel Regression and Wide Neural Networks”. In: *International Conference on Machine Learning, ICML*. 2020.
- [BCP24] Blake Bordelon, Hamza Tahir Chaudhry, and Cengiz Pehlevan. “Infinite Limits of Multi-head Transformer Dynamics”. In: *Advances in Neural Information Processing Systems, NeurIPS*. 2024.
- [BCV13] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. “Representation Learning: A Review and New Perspectives”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2013).
- [Bel21] Mikhail Belkin. “Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation”. In: *Acta Numerica* (2021).
- [BH89] Pierre Baldi and Kurt Hornik. “Neural networks and principal component analysis: Learning from examples without local minima”. In: *Neural Networks* (1989).
- [BKM23] Simone Bombari, Shayan Kiyani, and Marco Mondelli. “Beyond the Universal Law of Robustness: Sharper Laws for Random Features and Neural Tangent Kernels”. In: *International Conference on Machine Learning, ICML*. 2023.
- [BL22] Randall Balestriero and Yann LeCun. “Contrastive and Non-Contrastive Self-Supervised Learning Recover Global and Local Spectral Embedding Methods”. In: *NeurIPS*. 2022.
- [BL24] Randall Balestriero and Yann LeCun. “Learning by reconstruction produces uninformative features for perception”. In: *ArXiv preprint 2402.11337* (2024).
- [BNN22] Han Bao, Yoshihiro Nagano, and Kento Nozawa. “On the surrogate gap between contrastive and supervised losses”. In: *International Conference on Machine Learning*. 2022.
- [Bom+21] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. “On the Opportunities and Risks of Foundation Models”. In: *ArXiv preprint 2108.07258* (2021).
- [BPL22] Adrien Bardes, Jean Ponce, and Yann LeCun. “VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning”. In: *International Conference on Learning Representations, ICLR*. 2022.
- [Bro+93] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. “Signature Verification Using a Siamese Time Delay Neural Network”.



- In: *Advances in Neural Information Processing Systems*. Morgan Kaufmann, 1993.
- [Cab+23] Vivien Cabannes, Bobak Toussi Kiani, Randall Balestriero, Yann LeCun, and Alberto Bietti. “The SSL Interplay: Augmentations, Inductive Bias, and Generalization”. In: *International Conference on Machine Learning, ICML*. 2023.
- [Car+21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. “Emerging Properties in Self-Supervised Vision Transformers”. In: *IEEE/CVF International Conference on Computer Vision, ICCV*. 2021.
- [CB20] Lénaïc Chizat and Francis R. Bach. “Implicit Bias of Gradient Descent for Wide Two-layer Neural Networks Trained with the Logistic Loss”. In: *Conference on Learning Theory, COLT*. 2020.
- [Che+20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *International Conference on Machine Learning*. 2020.
- [Che+21] Yilan Chen, Wei Huang, Lam M. Nguyen, and Tsui-Wei Weng. “On the Equivalence between Neural Network and Support Vector Machine”. In: *Advances in Neural Information Processing Systems, NeurIPS*. 2021.
- [Den+24] Yuyang Deng, Junyuan Hong, Jiayu Zhou, and Mehrdad Mahdavi. “On the Generalization Ability of Unsupervised Pretraining”. In: *International Conference on Artificial Intelligence and Statistics*. 2024.
- [Dev+19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*. 2019.
- [DFH17] Lee H. Dicker, Dean P. Foster, and Daniel Hsu. “Kernel ridge vs. principal component regression: Minimax bounds and the qualification of regularization operators”. In: *Electronic Journal of Statistics* (2017).
- [DHP21] Ronald A. DeVore, Boris Hanin, and Guergana Petrova. “Neural network approximation”. In: *Acta Numerica* (2021).
- [DM20] Amit Daniely and Eran Malach. “Learning Parities with Neural Networks”. In: *Neural Information Processing Systems*. 2020.
- [EFG24] Pascal Mattia Esser, Maximilian Fleissner, and Debarghya Ghoshdastidar. “Non-parametric Representation Learning with Kernels”. In: *AAAI Conference on Artificial Intelligence*. 2024.
- [Ess+23] Pascal Mattia Esser, Satyaki Mukherjee, Mahalakshmi Sabanayagam, and Debarghya Ghoshdastidar. “Improved Representation Learning Through Tensorized Autoencoders”. In: *International Conference on Artificial Intelligence and Statistics, AISTATS*. 2023.
- [FAG25] Maximilian Fleissner, Gautham Govind Anil, and Debarghya Ghoshdastidar. “Infinite Width Limits of Self Supervised Neural Networks”. In: *The 28th International Conference on Artificial Intelligence and Statistics*. 2025. URL: <https://openreview.net/forum?id=CTlnhuuiaP>.

- [FFG24] Shlomo Libo Feigin, Maximilian Fleissner, and Debarghya Ghoshdastidar. “A Theoretical Characterization of Optimal Data Augmentations in Self-Supervised Learning”. In: *arXiv preprint arXiv:2411.01767* (2024).
- [FT74] Jerome H Friedman and John W Tukey. “A projection pursuit algorithm for exploratory data analysis”. In: *IEEE Transactions on computers* (1974).
- [Fu+23] Hengyu Fu, Tianyu Guo, Yu Bai, and Song Mei. “What can a Single Attention Layer Learn? A Study Through the RandomFeatures Lens”. In: *Advances in Neural Information Processing Systems, NeurIPS*. 2023.
- [Fu+24] Shi Fu, Yuzhu Chen, Yingjie Wang, and Dacheng Tao. “On Championing Foundation Models: From Explainability to Interpretability”. In: *ArXiv preprint 2410.11444* (2024).
- [Fuk98] Kenji Fukumizu. “Effect of Batch Learning in Multilayer Neural Networks”. In: *Advances in Neural Information Processing Systems, NeurIPS*. 1998.
- [GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [Ge+24] Jiawei Ge, Shange Tang, Jianqing Fan, and Chi Jin. “On the Provable Advantage of Unsupervised Pretraining”. In: *International Conference on Learning Representations*. 2024.
- [Gir21] Christophe Giraud. *Introduction to high-dimensional statistics*. CRC Press, 2021.
- [GWF14] Kristen B. Gorman, Tony D. Williams, and William R. Fraser. “Ecological Sexual Dimorphism and Environmental Variability within a Community of Antarctic Penguins (Genus *Pygoscelis*)”. In: *PLOS ONE* (2014).
- [Hao+21] Jeff Z. HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. “Provable Guarantees for Self-Supervised Deep Learning with Spectral Contrastive Loss”. In: *Advances in neural information processing systems*. 2021.
- [He+22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. “Masked Autoencoders Are Scalable Vision Learners”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*. 2022.
- [Hen+21] Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. “Unsolved problems in ml safety”. In: *ArXiv preprint 2109.13916* (2021).
- [HFG25] Jonghyun Ham, Maximilian Fleissner, and Debarghya Ghoshdastidar. “Impact of Bottleneck Layers and Skip Connections on the Generalization of Linear Denoising Autoencoders”. In: *CoRR* abs/2505.24668 (2025).
- [HG25] Yaomengxi Han and Debarghya Ghoshdastidar. “Attention Learning is Needed to Efficiently Learn Parity Function”. In: *CoRR* abs/2502.07553 (2025).
- [Hof+22] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. “Training compute-optimal large language models”. In: *Advances in Neural Information Processing Systems*. 2022.

- [Hol16] Gerhard B. Holt. “Potential Simpson’s Paradox in Multicenter Study of Intraperitoneal Chemotherapy for Ovarian Cancer”. In: *Journal of Clinical Oncology* (2016).
- [Hua+23] Shih-Cheng Huang, Anuj Pareek, Malte Jensen, Matthew P. Lungren, Serena Yeung, and Akshay S. Chaudhari. “Self-supervised learning for medical image classification: a systematic review and implementation guidelines”. In: *npj Digital Medicine* (2023).
- [HY21] Reinhard Heckel and Fatih Furkan Yilmaz. “Early Stopping in Deep Networks: Double Descent and How to Eliminate it”. In: *International Conference on Learning Representations, ICLR*. 2021.
- [Ize08] Alan J Izenman. *Modern multivariate statistical techniques*. Springer, 2008.
- [JHG18] Arthur Jacot, Clément Hongler, and Franck Gabriel. “Neural Tangent Kernel: Convergence and Generalization in Neural Networks”. In: *Advances in Neural Information Processing Systems, NeurIPS*. 2018.
- [JHM23] Daniel D. Johnson, Ayoub El Hanchi, and Chris J. Maddison. “Contrastive Learning Can Find An Optimal Basis For Approximately View-Invariant Functions”. In: *International Conference on Learning Representations*. 2023.
- [Kar+21] Stefani Karp, Ezra Winston, Yuanzhi Li, and Aarti Singh. “Local Signal Adaptivity: Provable Feature Learning in Neural Networks Beyond Kernels”. In: *Advances in Neural Information Processing Systems, NeurIPS*. 2021.
- [KB09] Tamara G Kolda and Brett W Bader. “Tensor decompositions and applications”. In: *Society for Industrial and Applied Mathematics, SIAM review* (2009).
- [Kra91] Mark A. Kramer. “Nonlinear principal component analysis using autoassociative neural networks”. In: *AICHE Journal* (1991).
- [KSS24] Chinmaya Kausik, Kashvi Srivastava, and Rishi Sonthalia. “Double Descent and Overfitting under Noisy Inputs and Distribution Shift for Linear Denoisers”. In: *Trans. Mach. Learn. Res.* 2024 (2024).
- [Kun+19] Daniel Kunin, Jonathan Bloom, Aleksandrina Goeva, and Cotton Seed. “Loss Landscapes of Regularized Linear Autoencoders”. In: *International Conference on Machine Learning, ICML*. 2019.
- [Lee+11] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. “Unsupervised learning of hierarchical representations with convolutional deep belief networks”. In: *Communications of the ACM* (2011).
- [LZB20] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. “On the linearity of large non-linear models: when and why the tangent kernel is constant”. In: *Advances in Neural Information Processing Systems*. 2020.
- [MM22] Song Mei and Andrea Montanari. “The Generalization Error of Random Features Regression: Precise Asymptotics and the Double Descent Curve”. In: *Communications on Pure and Applied Mathematics* (2022).
- [Mur22] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

- [NGG20] Kento Nozawa, Pascal Germain, and Benjamin Guedj. “PAC-Bayesian contrastive unsupervised representation learning”. In: *Conference on Uncertainty in Artificial Intelligence*. 2020.
- [Ope23] OpenAI. “GPT-4 Technical Report”. In: *ArXiv preprint 2303.08774* (2023).
- [PG20] Senthil Purushwalkam and Abhinav Gupta. “Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases”. In: *Advances in Neural Information Processing Systems, NeurIPS* (2020).
- [Ran+06] Marc’Aurelio Ranzato, Christopher S. Poultney, Sumit Chopra, and Yann LeCun. “Efficient Learning of Sparse Representations with an Energy-Based Model”. In: *Advances in Neural Information Processing Systems, NeurIPS*. 2006.
- [RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical image computing and computer-assisted intervention, MICCAI*. Springer. 2015.
- [RG22] Maria Refinetti and Sebastian Goldt. “The dynamics of representation learning in shallow, non-linear autoencoders”. In: *International Conference on Machine Learning, ICML*. 2022.
- [Sab+25] Mahalakshmi Sabanayagam, Lukas Gosch, Stephan Günnemann, and Debarghya Ghoshdastidar. “Exact Certification of (Graph) Neural Networks Against Label Poisoning”. In: *International Conference on Learning Representations, ICLR*. 2025.
- [SAE24] Mahalakshmi Sabanayagam, Omar Al-Dabooni, and Pascal Esser. “Cluster Specific Representation Learning”. In: *ArXiv preprint 2412.03471* (2024).
- [Sau+19] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. “A Theoretical Analysis of Contrastive Unsupervised Representation Learning”. In: *International Conference on Machine Learning, ICML*. 2019.
- [SH02] Peter Sollich and Anason S. Halees. “Learning Curves for Gaussian Process Regression: Approximations and Bounds”. In: *Neural Computation* (2002).
- [Sim+23] James B. Simon, Maksis Knutins, Ziyin Liu, Daniel Geisz, Abraham J. Fetterman, and Joshua Albrecht. “On the Stepwise Nature of Self-Supervised Learning”. In: *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Hawaii, USA*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. PMLR, 2023, pp. 31852–31876.
- [Sim51] E. H. Simpson. “The Interpretation of Interaction in Contingency Tables”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1951).
- [SMG14] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. “Exact solutions to the nonlinear dynamics of learning in deep linear neural networks”. In: *International Conference on Learning Representations, ICLR*. 2014.
- [SN23] Rishi Sonthalia and Raj Rao Nadakuditi. “Training Data Size Induced Double Descent For Denoising Feedforward Neural Networks and the Role of Training Noise”. In: *Transactions on Machine Learning Research, TMLR* (2023).

- [Sou+18] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. “The implicit bias of gradient descent on separable data”. In: *Journal of Machine Learning Research, JMLR* (2018).
- [SS02] Bernhard Schölkopf and Alexander Johannes Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning series. MIT Press, 2002.
- [SSM98] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. “Nonlinear component analysis as a kernel eigenvalue problem”. In: *Neural computation* (1998).
- [SWL23] Zhenmei Shi, Junyi Wei, and Yingyu Liang. “Provable Guarantees for Neural Networks via Gradient Feature Learning”. In: *Advances in Neural Information Processing Systems, NeurIPS*. 2023.
- [Tia22] Yuandong Tian. “Understanding Deep Contrastive Learning via Coordinate-wise Optimization”. In: *NeurIPS*. 2022.
- [TSL00] Joshua B Tenenbaum, Vin de Silva, and John C Langford. “A global geometric framework for nonlinear dimensionality reduction”. In: *Science* (2000).
- [UT19] Madeleine Udell and Alex Townsend. “Why Are Big Data Matrices Approximately Low Rank?” In: *SIAM Journal on Mathematics of Data Science* (2019).
- [vG24] Anna van Elst and Debarghya Ghoshdastidar. “Tight PAC-Bayesian risk certificates for contrastive learning”. In: *Under review at The SIAM Journal on Data Science (SIMODS)* (2024). URL: <https://arxiv.org/abs/2412.03486>.
- [VH08] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* (2008).
- [Vin+10a] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion”. In: *Journal of Machine Learning Research, JMLR* (2010).
- [Vin+10b] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.” In: *Journal of Machine Learning Research, JMLR* (2010).
- [Wag82] Clifford H. Wagner. “Simpson’s Paradox in Real Life”. In: *The American Statistician* (1982).
- [WL21] Zixin Wen and Yuanzhi Li. “Toward Understanding the Feature Learning Process of Self-supervised Contrastive Learning”. In: *International Conference on Machine Learning*. 2021.
- [WL22] Zixin Wen and Yuanzhi Li. “The Mechanism of Prediction Head in Non-contrastive Self-supervised Learning”. In: *NeurIPS*. 2022.
- [Yan+17] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. “Towards k-means-friendly spaces: Simultaneous deep learning and clustering”. In: *International Conference on Machine Learning, ICML*. 2017.

- [YH21] Greg Yang and Edward J. Hu. “Tensor Programs IV: Feature Learning in Infinite-Width Neural Networks”. In: *International Conference on Machine Learning, ICML*. 2021.
- [YL21] Greg Yang and Etai Littwin. “Tensor Programs IIb: Architectural Universality Of Neural Tangent Kernel Training Dynamics”. In: *International Conference on Machine Learning*, 2021.
- [Yoo+20] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. “VIME: Extending the Success of Self- and Semi-supervised Learning to Tabular Domain”. In: *Advances in Neural Information Processing Systems*. 2020.
- [Zbo+21] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. “Barlow Twins: Self-Supervised Learning via Redundancy Reduction”. In: *International Conference on Machine Learning, ICML*. 2021.
- [Zha+21] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. “Understanding deep learning (still) requires rethinking generalization”. In: *Communications of the ACM* (2021).