Architectural Inductive Biases and Moment Alignment: Analyzing Representational Capacity in Self-Supervised Learning Methods

Dontr (Dante) Lokitiyakul
University of Pennsylvania
dantehl@seas.upenn.edu

December 17, 2024

Abstract

Through analysis and empirical simulations on synthetic data, we investigate how the inductive biases of simple embedding architectures affect feature learning in a self-supervised learning method, MatrixSSL. We compare this to Spectral Contrastive Learning (SCL), a method with prior inductive bias aware analyses. Analogously to the mean-based condition for SCL from prior work, we find a second-order moment-based condition on embedding functions that guarantees optimal MatrixSSL alignment loss, producing a closed-form expression. Then provided a choice of embedding architecture and data distribution, we analyze the features are learnt in the embedding weights. We do so for several toy distributions, finding that in the case of linear and one layer ReLU network embeddings, neither method appears to have an advantage in learning features. Noticing then that this second-order moment-based assumption is equivalent to the mean-based assumption that minimizes SCL loss under a quadratic feature map, we propose that the representational benefits of the MatrixSSL loss can be achieved with a single quadratic transformation.

1 Introduction

Self-supervised learning (SSL) aims to replicate the capabilities of supervised training without labels, relying instead on structure inherent within data to learn effective representations without being bottlenecked by the cost of obtaining labeled data. Given the unparalleled availability of unlabeled data, SSL methods have served as a key component behind the empirical successes of current deep learning methods. Despite this, our theoretical understanding of SSL methods is still developing, and the forefront of SSL theory research has focused on contrastive learning methods. This thesis aims to expand upon current theoretical understanding of the interactions between self-supervised learning methods, architectural inductive biases, and their representational capacities, and how they affect downstream representation quality. In Section 2, we survey the dominant SSL methods in practive (Section 2.1) and provide a brief history of developments in SSL theory (Section 2.2). In Section 3, we hone in on Spectral Contrastive Learning, provide background on its theoretical development. We introduce a novel method, Matrix-SSL (Zhang et al. [2024]), which claims to provide a larger solution space of representations by aligning covariances of embeddings, rather than aligning embeddings. In Section 4 we perform various preliminary experiments and analyses to compare SCL and MatrixSSL. Section 5 builds upon the findings of Section 4, where we prove equivalence of second-order and first-order moment alignment losses up to a quadratic reparameterization. Our key contributions are as follows:

Contributions:

- We survey the landscape of self-supervised learning methods, and existing attempts to theoretically characterize it.

- We construct an alternative alignment loss to Matrix-SSL which aligns second-order moment information of embeddings. We find a second-order moment based condition on the embedding function (and data distribution), analogously to the first-order/mean-based condition in HaoChen and Ma [2022], which minimizes this second-order alignment loss.

- We find, through analysis and empirical simulations on synthetic data, that neither SCL nor MatrixSSL appear to outperform one another.

- **We prove that the first-order moment alignment loss in SCL is equivalent to the second-order moment alignment loss, up to a quadratic reparameterization.**

- We propose that the full SCL loss can, with the added quadratic transformation, in principle achieve the same representations that a second-order moment based method like Matrix-SSL can, and that any benefits of second-order methods are purely optimization related.

2 Survey of Self-Supervised Learning

Before we dive deeper, it helps to recall the goal of SSL: to obtain useful representations from unlabelled data. In the context of image classification, a widely adopted standard in SSL research, representations are broadly useful if 1) within the same class, representations are similar and invariant to spurious patterns within the data, and 2) representations are informative across classes, i.e representations of different classes are distinguishable. Of these, the first criterion is usually easier to implement explicitly. Since the data are unlabeled, a critical design question concerns how we define what images are within the same class. A common method in practice is use to use augmentations that preserve the semantics of the image (color distortions, rotations, crops, flips, etc.) to produce two views of the same image. These views are then treated as 'positive samples' from the same class, whose embeddings are encouraged to be similar by means of some combination of architecture and losses. However, a method that naively optimizes the first criterion only is prone to the *representation collapse problem*, in which all points in all classes are given the same constant embedding - in this case, criterion 1 is perfectly fulfilled, as the representations of all points in the same class are exactly the same, but criterion 2 fails completely, as the representation is unable to distinguish between different classes. A common trend amongst methods that explicitly optimize the first criterion is to implcitly prevent collapse through some implementation that ensures that different classes are embedded differently, and some methods covered below can be distinguished in this sense.

With that, most modern SSL methods can be broadly categorized into one of the following: contrastive learning , self-distillation based methods, CCA based methods, and masked image modeling (MIM) methods. We cover each below:

2.1 Methods in Practice

Contrastive Learning (CL) Contrastive learning (CL) methods aim to learn an embedding function such that similar points are embedded close to each other, whereas dissimilar points are embedded far apart within the embedding space. This is facilitated via a metric loss function defined over the embedding space. **SimCLR** (Chen et al. [2020]) embeds and projects two randomly augmented views of the same image through the same encoder and projector, and evaluates the quality of embeddings using a loss calculated over all pairs of augmentations: Denoting P as the set of positive pairs of projected embeddings (z_i, z_j) without duplicates by tuple ordering, they use the loss function

$$\mathcal{L}_{\text{NT-Xent}} = \frac{1}{2N} \sum_{(i,j \in P)} l_{i,j} + l_{j,i} \qquad \text{where} \quad l_{i,j} = -\log \left(\frac{\exp(\text{CoSim}(z_i, z_j)/\tau}{\sum_{k=1,k\neq i}^{2N} \exp(\text{CoSim}(z_i, z_k)/\tau} \right)$$

Where CoSim(., .) is the cosine similarity between two vectors (i.e the normalized dot product). Note that the loss is minimized by maximizing similarity between positive pairs, and minimizing similarity between augmentations of different images (negative pairs). SimCLR, like other CL methods, avoids the collapse problem by explicitly incorporating the similarity of negative pairs in the loss function. While SimCLR can perform very well empirically, the method is primarily disdvantaged by the need for many negative samples to perform effective CL, which can result in large training times. Another CL method, **MoCo** (He et al. [2020]), reduces the large batch size requirement of SimCLR by using two different encoders, an online and a momentum encoder. The weights of the online network are updated via backpropagation, and the momentum encoder's weights are updated as an exponential moving average of the online encoder's weights. The outputs of these networks are fed into a contrastive loss as before, but MoCo uses a moving dictionary of updated embeddings in place of a large batch, resuing embeddings from earlier batches.

Self-Distillation Based Methods Self-Distillation Based methods are built on the idea that, given embeddings of two augmented views of the same image, one can improve representations by trying to predict one embedding from the other, and optimizing so that the prediction is similar to the target embedding. On its own, this would lead to collapsed representations, so in practice this is avoided by incorporating some implementational structure. The most well-known of these, **BYOL** (Grill et al. [2020]), uses two separately parameterized encoder-projectors, an online network whose output is used to predict the output of a target network. Only the weights of the online encoder are updated via backpropagation, while the target network's weights are set to be an exponential average of the online network, so that both encoders' representations improve with each iteration. This asymmetry is crucial to BYOL's performance.

Canonical-Correlation Analysis (CCA) Based Methods Canonical Correlation Analysis finds a linear transformation of two views of data such that the dimensions of the transformed data are uncorrelated. CCA-based methods attempt to decorrelate the representations of two different views.

Masked Image Modeling Methods Masked Image Modeling (MIM) methods mask portions of an input image and attempt to generate them. MIM methods evolved as an image-domain equivalent to Masked Language Modeling (MLM) methods. MIM methods developed from BEiT Bao et al. [2021], which cast the existing classification problem of BERT Devlin et al. [2018], an MLM method, to a regression problem. The two main methods in this field are Masked Autoencoders (MAE) He et al. [2022] and SimMIM Xie et al. [2022].

2.2 Self-Supervised Learning Theory

While research in SSL theory is an emerging field, there exist various lines of study across various methods. This section outlines some developments.
One such line of work involves delineating precise conditions on which performance guarantees can be made from learned representations. Arora et al. [2019] do so for contrastive learning algorithms. Given positive pairs (x, x^+) that can be sampled conditionally independently (CI) given an underlying latent variable (such as classes), and negative samples x^- that can be sampled independently, then minimizing the

unsupservised contrastive loss function $L(f) = \mathbb{E}_{x,x^+,x^-}[f(x)^\top(f(x^+) - f(x^-)))]$ results in an embedding $\hat{f}$ on which a linear classifier achieves error bounded by the minimum unsupervised loss, plus a generalization error dependent on the Rademacher Complexity of the function class of f. However, the conditional independence assumption is unrealistic, as practically positive pairs are only independent conditioned on the class label (which is unavailable in SSL), or conditioned on the original image if using augmentations.

Similarly, Lee et al. [2021], focusing on reconstruction based methods that involve predicting some proxy variable X_2 from X_1 for downstream label Y, show that when X_1, X_2 are CI given Y, and $X_2|Y$ has rank equal to the number of classes, then optimizing an embedding to predict X_2 produces an embedding function $\psi^* = \arg\min_g \mathbb{E}\|X_2 - g(X_1)\|^2$ on which a linear classifier can perfectly replicate the true function. However, they also extend their analysis to Approximate CI (ACI) by defining a measure for ACI: $\epsilon_{\text{CI}}^2 = \mathbb{E}_{X_1}[\|\mathbb{E}[X_2|X_1] - \mathbb{E}_{Y,Z}[\mathbb{E}[X_2|Y,Z]|X_1]\|^2]$, where Z are further latent variables that can be conditioned on. They show that for large enough labelled and unlabelled samples, the generalization error of the linear classifier on the embeddings is bounded by a sum including ϵ_{CI}, error on the pretext task (to form the approximation error), and an estimation error that depends on the number of labelled samples. A closely related study is Tosh et al. [2021], which examines the multi-view setting, where X_1, X_2 are views of/with label Y. They show that, when X_1, X_2 contain redundant information about Y, then a strategy for predicting Y is to predict X_2 from X_1, then predict Y from the prediction of X_2, which they show has bounded error. Specifically, they show that

$$\mathbb{E}[(\mathbb{E}[\mathbb{E}[Y|X_2]|X_1] - \mathbb{E}[Y|X_1,X_2])^2] \leq (\sqrt{\varepsilon_{X_1}} + \sqrt{\varepsilon_{X_2}})^2$$

where $\varepsilon_W = \mathbb{E}[(\mathbb{E}[Y|W] - \mathbb{E}[Y|X_1,X_2])^2]$ (for $W = X_1$ or X_2) is a measure of the redundancy between X_1, X_2 wrt Y (the smaller, the more redundancy). The LHS of the inequality gives the error of the strategy outlined above.

HaoChen et al. [2021] develop a more realistic framework for contrastive learning without the conditional independence assumption of classes in Arora et al. [2019]. They consider an underlying *population augmentation graph* where any possible augmentation of a natural datum corresponds to a node, and edges connect augmentations that could have been produced from same natural data. The key idea is that, despite not having all nodes within the same class being directly connected by an edge (thus allowing the distribution of augmentations to change throughout the class), any two nodes within the same class are connected via a series of augmentations. At the same time, any two augmentations from different classes are highly unlikely to be connected by an edge. With this setup, a natural clustering structure corresponding to each class arises within the population augmentation graph, and the task of learning representations can be reframed as performing spectral clustering on the population augmentation graph. In traditional spectral graph theory, this involves eigendecomposing the Laplacian matrix and stacking the largest m eigenvectors as columns in an embedding matrix. The rows of this embedding matrix then serve as the embedding of the corresponding data in the graph. Since this does not provide a parametric embedding function that can be applied to other points, the authors parameterize the rows of the embedding matrix with the weights of a neural network, and minimize a loss that encourages the network weights to learn these feature extractors. They find that the loss that encourages is contrastive in nature - they term their task **Spectral Contrastive Learning** (SCL). For an embedding function f, the Spectral Contrastive Loss is

$$\mathcal{L}_{\text{SCL}} = \mathbb{E}_{(x,x')\sim p_+} \|f(x) - f(x')\|_2^2 + \lambda \|\mathbb{E}_{x\sim p_d}[f(x)f(x)^\top] - \mathbb{I}\|_F^2$$

3 Investigating SCL and Matrix-SSL

We now turn our focus to investigating the representational capacity of two specific self-supervised learning methods, Spectral Contrastive Learning (SCL) and Matrix-SSL.

3.1 Notation

Let (x, x') denote a positive pair, commonly viewed as two randomly sampled augmentations of the same natural image. The distribution over positive pairs p_+ is a symmetric distribution over $\mathcal{X} \times \mathcal{X}$, where $\mathcal{X} \subset \mathbb{R}^l$ denotes the space of possible augmentations. Marginalizing $p_+(\cdot,\ \cdot)$ over either entry results in the (same) marginal distribution over augmentations, which we denote p_d. We assume augmentation data is l-dimensional and embeddings are k-dimensional. Throughout we denote embedding functions as $f : \mathcal{X} \to \mathbb{R}^k$, from function class $\mathcal{F}$. We refer to the ith row and jth column of matrix Z as $(Z)_{i,\cdot}$ and $(Z)_{\cdot,j}$ respectively.

3.2 Background

Spectral Contrastive Learning (SCL) is a contrastive learning method, supported theoretically by the population augmentation graph construction which applies to more realistic settings. The SCL loss is written as

$$\mathcal{L}_{\text{SCL}} = \mathbb{E}_{(x,x')\sim p_+} \|f(x) - f(x')\|_2^2 + \lambda \|\mathbb{E}_{x\sim p_d}[f(x)f(x)^\top] - \mathbb{I}\|_F^2$$

Two important developments in SSL theory research should also be noted: First is the alignment and uniformity framework in contrastive learning developed by Wang and Isola [2022], which suggests that the successes of contrastive learning methods can be attributed to two properties of the loss function: alignment, which encourages closeness of positive pairs in embedding space, and uniformity, which encourages the distribution of (normalized) embeddings to be uniform on the hypersphere. Under this framework, we can interpret the SCL loss terms accordingly: the first term encourages alignment by incentivizing positive pairs to be similarly embedded, and the second term encourages uniformity by incentivizing the covariance matrix of embeddings to be close to the uniform distribution. Second is the need to account for the inductive biases of the architecture and training algorithm used to learn representations. In the context of contrastive learning, Saunshi et al. [2022] underscores this importance by providing a simple learning example on which different architectures yield vastly differing downstream performances.

In a follow-up to this and their original SCL paper, HaoChen and Ma [2022] consider assumptions on the inductive biases of the architecture required to guarantee downstream classification performance for SCL learnt representations. Specifically, for embedding function f they assume that (1) for any augmentation x, the embedding function f retains its value $f(x)$ on average over other points x' that could form a positive pair with x, and (2) the dimensions of the embeddings are orthogonal on average over all augmentations. Mathematically, these assumptions are

Assumption 1 $f(x) = \mathbb{E}_{x'|x}[f(x')]$

Assumption 2 $\mathbb{E}_{x\sim p_d}[f(x)f(x)^\top] = I_d$

where $p(x'|x) := p_+(x, x')/p_d(x)$ is the conditional distribution of augmentation x' given x. More formally, they assume the existence of approximately orthogonal eigenfunctions to the Laplacian operator over the population augmentation graph.

A recent study proposed **Matrix-SSL** (Zhang et al. [2024]), an SSL method claiming to achieve better downstream performance over existing methods. They specifically rewrite the alignment loss to encourage similarity in the covariance matrices of embeddings, rather than similarity of the embedding matrices themselves. The Matrix-SSL loss is defined with respect to sample covariances of embeddings calculated using batches. Let $X_1, X_2 \in \mathbb{R}^{n \times l}$ denote size n batches of augmentations, where the ith row of X_1 and X_2 form a positive pair. Denote the matrix of embeddings of these rows as $Z_1, Z_2 \in \mathbb{R}^{n \times k}$, where k is the embedding dimension (i.e $(Z_*)_{i,\cdot} = f((X_*)_{i,\cdot})$. The sample covariances are then denoted $C(Z_1, Z_2) := \frac{1}{n} Z_1^\top H_n Z_2$, where $H_n := I_n - \frac{1}{n} 1_n 1_n^\top$ is the idempotent centering matrix. Then, the Matrix-SSL loss is defined as

$$\mathcal{L}_{\text{Matrix-SSL}} = \text{MCE}\left(\frac{1}{k} I_k, C(Z_1, Z_2)\right) - \operatorname{Tr} C(Z_1, Z_2) + \gamma \text{MCE}(C(Z_1, Z_1), C(Z_2, Z_2))$$

Where $\text{MCE}(P, Q)$ is the Matrix Cross-Entropy of matrices P, Q, the matrix equivalent of cross-entropy. This quantity is minimized when $P = Q$. The first term corresponds to the uniformity loss, encouraging the sample covariance (across embeddings) to be equal to the identity, and the remaining terms encourage the sample autocovariances $C(Z_1, Z_1), C(Z_2, Z_2)$ to be similar. The authors claim this allows for a larger solution space of embeddings - intuitively, this makes sense, as it's possible to have $Z_1 Z_1^\top = Z_2 Z_2^\top$ (as the Matrix-SSL loss would encourage) but not have $Z_1 = Z_2$ (as typical contrastive learning methods encourage).

4 Preliminary Results

Second Order Moment Loss As before, the Matrix-SSL alignment term is minimized when the covariances of embeddings $C(Z_1, Z_1)$ and $C(Z_2, Z_2)$ are equal. We note this is an unusual method for aligning second-order information across augmentations, as this can be made true simply by making batch sizes large. Specifically, the condition $C(Z_1, Z_1) = C(Z_2 Z_2)$ is a finite sample version of the condition

$$\|\mathbb{E}_{(x,x')\sim p_+}[f(x)f(x)^\top - f(x')f(x')^\top]\|_F^2 = 0$$

But since the positive pair distribution is already symmetric, then the marginal distributions are equal, so

$$\begin{aligned}&\mathbb{E}_{(x,x')\sim p_+}[f(x)f(x)^\top - f(x')f(x')^\top]\\&= \mathbb{E}_{x\sim p_d}[f(x)f(x)^\top] - E_{x'\sim p_d}[f(x')f(x')^\top] = 0\end{aligned}$$

so the LHS itself already evaluates to 0. In the finite sample space, the condition can be written as

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n}\left(f(x_i) - \frac{1}{n}\sum_{i=1}^{n} f(x_i)\right)\left(f(x_i) - \frac{1}{n}\sum_{i=1}^{n} f(x_i)\right)^\top\\&= \frac{1}{n}\sum_{i=1}^{n}\left(f(x_i') - \frac{1}{n}\sum_{i=1}^{n} f(x_i')\right)\left(f(x_i') - \frac{1}{n}\sum_{i=1}^{n} f(x_i')\right)^\top\end{aligned}$$

and as n approaches infinity, the two quantities can approach each other. Note that this holds regardless of the choice of f, so in principle the Matrix-SSL loss could be minimized without learning any relevant features (eg. if f were the identity mapping). Given this and the added difficulty of analyzing matrix logarithm terms, we consider an alternative loss that still aligns second-order moment information while avoiding this property inherent in Matrix-SSL. Specifically, we define the second order alignment term

$$\mathcal{L}_{\text{MeanNormDiff}} = \mathbb{E}_{(x,x')\sim p_+}\|f(x)f(x)^\top - f(x')f(x')^\top\|_F^2$$

and and consider the loss

$$\mathcal{L}_{\text{SecondOrder}}(f) = \mathcal{L}_{\text{MeanNormDiff}} + \|\mathbb{E}_{x\sim p_d}[f(x)f(x)^\top] - \mathbb{I}\|_F^2$$

For the alignment term in this loss, the expectation is effectively taken outside the norm, rather than within. From here onwards, whenever we mathematically analyze loss terms, we consider this second order loss in place of Matrix-SSL.

Next, we show that, when we consider a second-order moment-based assumption analogous to Assumption 1, the alignment loss $\mathcal{L}_{\text{MeanNormDiff}}$ is minimized.

Assumption 3 $f(x)f(x)^\top = \mathbb{E}[f(x')f(x')^\top|x], \quad \forall x \in \mathcal{X}.$

Note that this isn't exactly an assumption on the conditional covariances, but rather an assumption on the conditional second-order moment of embeddings. Under this assumption, the alignment term $\mathcal{L}_{\text{MeanNormDiff}}(f)$ is minimized:

Theorem 1 *Assumption 3 minimizes the second-order moment alignment term $\mathcal{L}_{MeanNormDiff}$*

Proof: We start by expanding the loss:

$$\begin{aligned}\mathcal{L}_{\text{MeanNormDiff}}(f) &= \mathbb{E}_{(x,x')\sim p_+}\|f(x)f(x)^\top - f(x')f(x')^\top\|_F^2 \\ &= 2\mathbb{E}_{x\sim p_d}\|f(x)f(x)^\top\|_F^2 - 2\mathbb{E}_{(x,x')\sim p_+}\Big[\text{Tr}(f(x)f(x)^\top f(x')f(x')^\top)\Big]\end{aligned}$$

Then, looking at the trace term, we get

$$\begin{aligned}&-2\mathbb{E}_{(x,x')\sim p_+}\Big[\text{Tr}(f(x)f(x)^\top f(x')f(x')^\top)\Big] \\ &= -2\mathbb{E}_{x\sim p_d}\Big[\text{Tr}(f(x)f(x)^\top \mathbb{E}_{x'|x}[f(x')f(x')^\top])\Big] && \text{applying the law of iterated expectation} \\ &= -2\mathbb{E}_{x\sim p_d}\Big[\text{Tr}(f(x)f(x)^\top f(x)f(x)^\top])\Big] && \text{applying Assumption 3} \\ &= -2\mathbb{E}_{x\sim p_d}\|f(x)f(x)^\top\|_F^2 && \text{since}\,\text{Tr}(AA^\top) = \|A\|_F^2\end{aligned}$$

Substituting this back into the loss, we end up with $\mathcal{L}_{\text{MeanNormDiff}}(f) = 0$. Since the Frobenius norm is always non-negative, then the term is minimized with value 0.

Analysis on Synthetic Data We now look for synthetic self-supervised learning tasks, consisting of an augmentation scheme and a (downstream) labeling function, for which representations learnt by MatrixSSL perform well on downstream tasks, while representations learnt by SCL perform poorly. We do so primarily via mathematical analysis (using the second order loss in place of Matrix-SSL). Since we note that Assumptions 1 and 3 each minimize their respective alignment losses, then given assumptions on the embedding function class of f and the positive pair distribution p_+, the Assumptions provide a closed form expression on which we can analyze what the model weights can learn. Drawing from Saunshi et al. [2022], we focus primarily on linear embedding functions $f(x) = Wx$ in our analyses.

One set up was in $\mathbb{R}^3$, where augmentations had the form (x_1, x_2, x_3) and $(-x_1, -x_2, x_3)$. Under Assumption 1, SCL would learn to ignore the first 2 dimensions (i.e learn zero weights for these dimensions), and learn 'free' weights for the third dimension. If the labels were determined by $y = \text{sign}(x_3)$, then SCL could properly learn the task. Instead, using a label given by $\text{sign}(x_1 x_2)$ would not only preserve the label across augmentations, but also lead to poor downstream performance for SCL. On the other hand, for a 3×3 embedding function W, simply setting $W_{i3} = 0$ for all $i \in [3]$ would satisfy the corresponding assumption in MatrixSSL:

$$f(x)f(x)^\top - \mathbb{E}_{x'|x}[f(x')f(x')^\top] = W(xx^\top - x'x'^\top)W^\top$$

$$= \begin{bmatrix} w_{11} & w_{12} & 0 \\ w_{21} & w_{22} & 0 \\ w_{31} & w_{32} & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 2x_1x_3 \\ 0 & 0 & 2x_2x_3 \\ 2x_1x_3 & 2x_2x_3 & 0 \end{bmatrix} \begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{0}$$

The resulting representation would ignore the third dimension and could properly assign weights for the first two dimensions. However, the labeling function itself is quadratic in the inputs, and so a quadratic classifier would be needed.

Another initial example we tried was to directly use the example from Saunshi et al. [2022]: d dimensional boolean hypercube data for natural data, with augmentations performed by leaving the first k dimensions unchanged and scaling each of the last $d-k$ dimensions by some factor τ distributed uniformly in $(0, 1]$, sampled i.i.d across both dimensions and augmentation sets. For brevity we call this the 'multiplication' (mult.) augmentation scheme, and we analyze this in with $d = 3$, $k = 1$. Since the natural data is boolean, then given an augmentation one can actually recover exactly the natural data that generated it (one need only look at the sign of the augmented dimension). Further, since the augmentations involve independent scaling of the last $d-k$ dimensions, the only labeling function that preserves labels across augmentations is one that depends on the k unaugmented dimensions. Then we can see that for the assumption in SCL to hold, we have that

$$g_i(x) - \mathbb{E}_{x'|x}[g_i(x')] = w^\top(0, (\tau_2 - 0.5)x_2, (\tau_3 - 0.5)x_3) = 0$$

where w is the ith row of embedding function matrix W. This then learns free weights for the (first) unaugmented dimension, and zero weights for the augmented dimensions, since $w_2(\tau_2 - 0.5)x_2) + w_3(\tau_3 - 0.5)x_3)$ must be equal to 0 for any choice of $x_2, x_3 \in \pm 1$, and any choice of $\tau_2, \tau_3 \in (0, 1]$. Thus, SCL does well on the hypercube task, suggesting we need to modify the setup slightly.

Since intuitively MatrixSSL learns to match the autocovariance matrices of the embeddings across different augmentations, whereas SCL learns to directly match the embedding matrices, the next tasks we tried involved correlating the dimensions of the data together. We considered two such augmentation schemes, both with the same natural boolean hypercube data setting as earlier: first we consider pairs of dimensions in the last $d-k$ dimensions, and sample a single τ uniformly in $[-1,1]$ for each; we then add τ to one of these dimensions, and subtract it from the other. This allows us to have the label depend on either only the augmented dimensions or on only the augmented dimensions, without the augmentations changing the label. We call this the 'addition' augmentation scheme. The assumptions on SCL predicts that it learns free weights on unaugmented dimensions, and equal weights for each pair of augmented dimensions.

In the other, we sample τ the same way for each pair of augmented dimensions, but we add to each augmented dimension the quantity τ multiplied by the other augmented dimension, i.e $x_2 + \tau x_3$, and $x_3 + \tau x_2$ in our example in $\mathbb{R}^3$. The SCL assumptions predicts that it again learns free weights for unaugmented dimensions, and can have either equal or negated weights for each pair of augmented dimensions.

Another setting we consider is one where data points x are constructed as a concatenation of k 'features', each with dimension d: $x = (x^{(1)}, ...x^{(k)})$, where features are drawn from a marginal normal distribution, but across positive pairs, the features are correlated along some underlying feature direction v_j: for each $j \in [k]$:

$$(x^{(j)}, x^{(j)\prime}) \sim \mathcal{N}\left(0_{2d}, \begin{bmatrix} I_d & v_j v_j^\top \\ v_j v_j^\top & I_d \end{bmatrix}\right)$$

The existence of latent features allows us to empirically check whether our embeddings have properly learnt features. We term this data/augmentation setting 'correlated normal features'. Through analyses, we find effectively the same solutions for SCL and the second order loss:

$$W_{\text{SCL}} = \begin{bmatrix} \frac{v_1}{\|v_1\|}^\top & 0_d^\top \\ 0_d^\top & \frac{v_2}{\|v_2\|}^\top \end{bmatrix} \quad \text{and} \quad W_{\text{SecondOrder}} = \begin{bmatrix} \frac{v_1}{\sqrt{2}\|v_1\|}^\top & 0_d^\top \\ 0_d^\top & \frac{v_2}{\sqrt{2}\|v_2\|}^\top \end{bmatrix}$$

Empirically, we train linear embeddings using Matrix-SSL and SCL on correlated normal data with 5 features, each with dimension 5 (with underlying feature vectors sampled randomly), setting the embedding dimension to 5. We evaluate the embeddings by generating data according to the normal distribution, and for each feature, labeling them by $y = \text{sign}(v_j^\top x^{(j)})$. Then we train a linear probe on top of the embeddings and evaluate accuracies for each feature, shown in Tables 1 and 2, and visualized in Figure 1. Visualizations are also shown in Figure 2 for the same training setup but with a ReLU added onto the embedding.

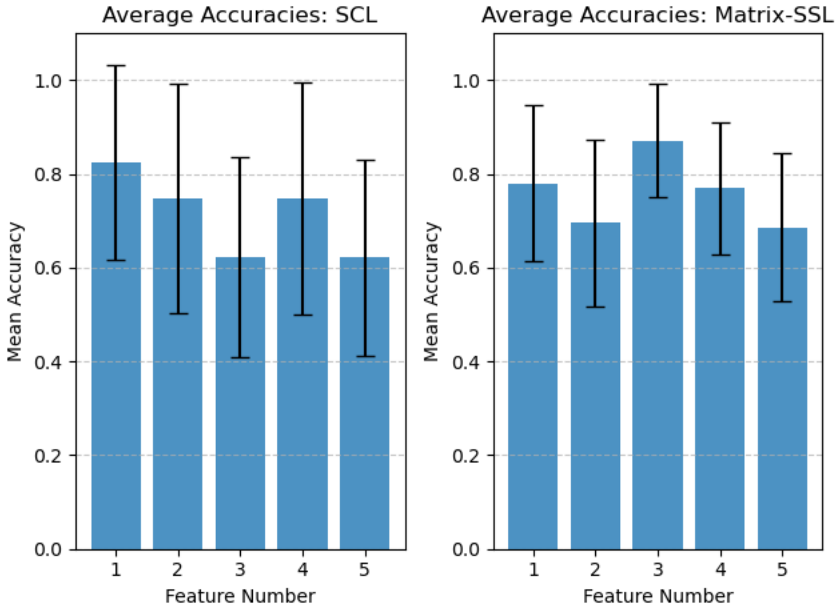

Figure 1: Average classification accuracies over five features for linear embeddings learned by SCL and Matrix-SSL respectively. Note how it isn't clear whether one method outperforms the other.

	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
Run 1	0.486	0.504	0.503	0.997	0.499
Run 2	0.993	0.499	0.496	0.504	0.984
Run 3	0.996	0.996	0.498	0.499	0.500
Run 4	0.821	0.992	0.992	0.993	0.503

Table 1: Classification accuracies for each of five features, across four training runs of SCL with a linear embedding architecture, trained on normal correlated feature data.

	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
Run 1	0.947	0.521	0.662	0.947	0.543
Run 2	0.708	0.951	0.929	0.861	0.568
Run 3	0.923	0.775	0.948	0.684	0.692
Run 4	0.540	0.539	0.947	0.590	0.941

Table 2: Classification accuracies for each of five features, across four training runs of Matrix-SSL with a linear embedding architecture, trained on normal correlated feature data.

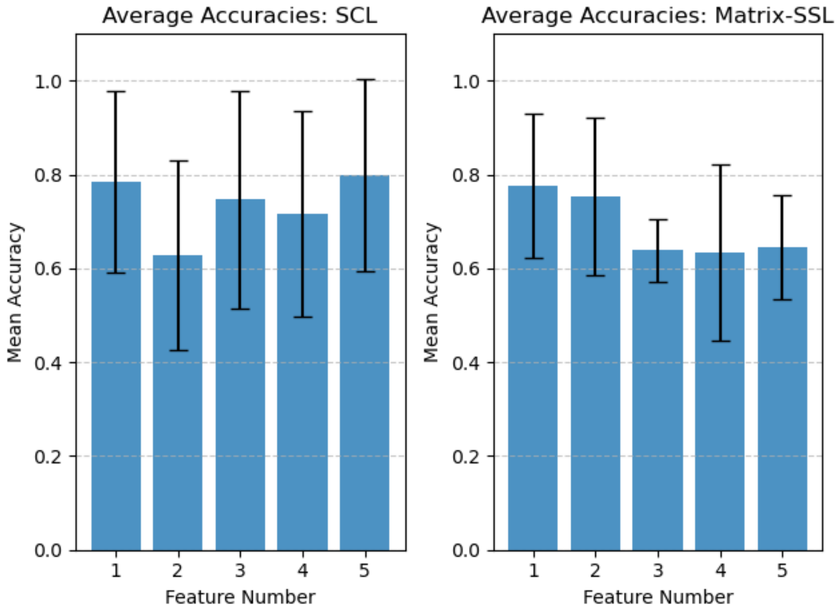

Figure 2: Average classification accuracies over five features for one-layer ReLU embeddings learned by SCL and Matrix-SSL respectively.

5 Analysis

We note that Assumption 3 is equivalent to Assumption 1 with an added fixed quadratic transformation. To give a minimal example, any function $f : \mathcal{X} \to \mathbb{R}^k$ satisfying Assumption 3 defines a function g, given by

$$g : \mathcal{X} \to \mathbb{R}^{k(k+1)/2}, \quad g(x) = (f(x)_1^2, ..., f(x)_k^2, f(x)_1 f(x)_2, ...f(x)_{k-1} f(x)_k)$$

where the entries of g are all possible quadratic products of the outputs of f. Then $f(x)f(x)^\top = \mathbb{E}[f(x')f(x')^\top | x]$ directly implies $g(x) = \mathbb{E}[g(x')|x]$. This suggests that, in terms of representational power, methods that align second-order information are no more powerful than those that align first-order moments given a quadratic transformation applied to its representations. This transformation can be fixed, as given by the construction g above, or approximated to a certain degree with a number of extra layers in the embedding function. We formalize this as follows:

Theorem 2 *Suppose f is an embedding function with $\mathcal{L}_{MeanNormDiff}(f) \leq \epsilon$. Then one can construct from f a new embedding function g such that $\mathcal{L}_{SCL}(g) \leq \epsilon$*

Proof: We first expand the loss, which we know is bounded above by ϵ:

$$\begin{aligned}
&\mathcal{L}_{\text{MeanNormDiff}}(f) = \mathbb{E}_{(x,x')\sim p_+} \| f(x)f(x)^\top - f(x')f(x')^\top \|_F^2 \\
&= \mathbb{E}_{(x,x')\sim p_+} \sum_{i,j \in [k]} (f(x)_i f(x)_j - f(x')_i f(x')_j)^2 \\
&= \sum_{i \in [k]} \mathbb{E}_{(x,x')} (f(x)_i^2 - f(x')_i^2)^2 + 2 \sum_{i<j \in [k]} \mathbb{E}_{(x,x')} (f(x)_i f(x)_j - f(x')_i f(x')_j)^2 \leq \epsilon
\end{aligned}$$

We will use the quadratic construction g from above. Then we can expand $\mathcal{L}_{\text{SCL}}(g)$:

$$\begin{aligned}
&\mathcal{L}_{\text{SCL}}(g) = \mathbb{E}_{(x,x')\sim p_+}\|g(x) - g(x')\|_2^2 \\
&= \mathbb{E}_{(x,x')\sim p_+}\sum_{i\in[k]}(g(x)_i - g(x')_i)^2 + \sum_{i=k+1}^{k(k+1)/2}(g(x)_i - g(x')_i)^2 \\
&= \sum_{i\in[k]}\mathbb{E}_{(x,x')}(f(x)_i^2 - f(x')_i^2)^2 + \sum_{i<j\in[k]}\mathbb{E}_{(x,x')}(f(x)_i f(x)_j - f(x')_i f(x')_j)^2
\end{aligned}$$

where we get the last line since the first k terms of g correspond to squared terms of f, and the remaining terms correspond to quadratic terms $f_i f_j$ where $i < j$. Since all the terms within the expectations are positive, we know the expectation itself must be positive. From comparing expanded terms, we see then that $\mathcal{L}_{\text{SCL}}(g) < \mathcal{L}_{\text{MeanNormDiff}}(f) \leq \epsilon$, so $\mathcal{L}_{\text{SCL}}(g) \leq \epsilon$.

In other words, we've shown in principle that for any function f with low second-order alignment loss, a quadratically transformed version of f would also achieve los first-order/SCL alignment loss.

However, we note that the alignment loss is only part of the entire self-supervised loss, and we must account for the uniformity loss as well. We leave proofs and empirical investigations to future work, but we conjecture that a similar construction could achieve low SCL as well:

Conjecture 1 *Suppose f is an embedding function with $\mathcal{L}_{SecondOrder}(f) \leq \epsilon$. Then one can construct from f a new embedding function g which satisfies $\mathcal{L}_{SCL} \leq h(\epsilon)$, where $h(\epsilon)$ is some function of ϵ.*

References

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

H. Bao, L. Dong, S. Piao, and F. Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

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

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

J. Z. HaoChen and T. Ma. A theoretical study of inductive biases in contrastive learning. *arXiv preprint arXiv:2211.14699*, 2022.

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

K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

J. D. Lee, Q. Lei, N. Saunshi, and J. Zhuo. Predicting what you already know helps: Provable self-supervised learning. *Advances in Neural Information Processing Systems*, 34:309–323, 2021.

N. Saunshi, J. Ash, S. Goel, D. Misra, C. Zhang, S. Arora, S. Kakade, and A. Krishnamurthy. Understanding contrastive learning requires incorporating inductive biases. In *International Conference on Machine Learning*, pages 19250–19286. PMLR, 2022.

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

T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere, 2022.

Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022.

Y. Zhang, Z. Tan, J. Yang, W. Huang, and Y. Yuan. Matrix information theory for self-supervised learning, 2024. URL `https://openreview.net/forum?id=e1IMBXiDhW`.