Matrix Information Theory for Self-Supervised Learning

Yifan Zhang [*1] **Zhiquan Tan** [*2] **Jingqin Yang** [*1] **Weiran Huang** [3 4] **Yang Yuan** [1 4 5]

Abstract

The maximum entropy encoding framework provides a unified perspective for many non-contrastive learning methods like SimSiam, Barlow Twins, and MEC. Inspired by this framework, we introduce Matrix-SSL, a novel approach that leverages matrix information theory to interpret the maximum entropy encoding loss as matrix uniformity loss. Furthermore, Matrix-SSL enhances the maximum entropy encoding method by seamlessly incorporating matrix alignment loss, directly aligning covariance matrices in different branches. Experimental results reveal that Matrix-SSL outperforms state-of-the-art methods on the ImageNet dataset under linear evaluation settings and on MS-COCO for transfer learning tasks. Specifically, when performing transfer learning tasks on MS-COCO, our method outperforms previous SOTA methods such as MoCo v2 and BYOL up to 3.3% with only 400 epochs compared to 800 epochs pre-training. We also try to introduce representation learning into the language modeling regime by fine-tuning a 7B model using matrix cross-entropy loss, with a margin of 3.1% on the GSM8K dataset over the standard cross-entropy loss.

1. Introduction

Contrastive learning methods (Chen et al., 2020a; He et al., 2020) focus on aligning similar objects closely while distancing dissimilar ones. This approach, grounded in intuitive principles, has led to deep and interesting insights. For example, SimCLR has been proved to perform spectral clustering on similarity graph (Tan et al., 2023b; HaoChen et al., 2021), and Wang & Isola (2020) highlight two critical aspects of contrastive loss: **alignment and uniformity**.

Alignment loss ensures similar objects are closely mapped, whereas uniformity loss promotes a uniformly distributed output feature space that preserves the maximum information. Remarkably, many existing contrastive methods (Wu et al., 2018; He et al., 2020; Logeswaran & Lee, 2018; Tian et al., 2020a; Hjelm et al., 2018; Bachman et al., 2019; Chen et al., 2020a) can be viewed as specific implementations of these two loss types, a perspective that simplifies understanding their core mechanisms.

Simultaneously, there is growing interest in non-contrastive learning methods that do not use negative samples, such as BYOL (Grill et al., 2020), SimSiam (Chen & He, 2021), Barlow Twins (Zbontar et al., 2021), VICReg (Bardes et al., 2021), etc. Among these, Liu et al. (2022) presented an interesting theoretical framework called maximum entropy encoding, which proposes to maximize the following loss between the two feature matrices $\mathbf{Z}_1, \mathbf{Z}_2$ computed from different augmentations from the same input:

$$\mathcal{L}_{\text{MEC}} = -\mu \log\det\left(\mathbf{I}_d + \lambda \mathbf{Z}_1 \mathbf{Z}_2^\top\right).$$

Although it may not be immediately obvious, the above loss encourages maximum entropy encoding for the feature embeddings, which is similar to the **uniformity loss** in contrastive learning methods. It turns out that this formulation naturally encompasses loss functions of several other non-contrastive methods like SimSiam, Barlow Twins, and the resulting algorithm MEC surpasses previous methods in performance (Liu et al., 2022) (element-wise alignment losses such as $\|\mathbf{z}_1 - \mathbf{z}_2\|_2$ used in BYOL can be seen as low-order Taylor expansion terms in this MEC loss). However, a comparison of contrastive and non-contrastive methods reveals some differences:

Learning Method	Loss Function
Contrastive Learning	Uniformity + Alignment
Non-contrastive Learning	Uniformity

This observation naturally propels us towards a broader, more explorative query:

[*]Equal contribution [1]IIIS, Tsinghua University, Beijing, China [2]Department of Mathematical Sciences, Tsinghua University, Beijing, China [3]MIFA Lab, Qing Yuan Research Institute, SEIEE, Shanghai Jiao Tong University, Shanghai, China [4]Shanghai AI Laboratory, Shanghai, China [5]Shanghai Qizhi Institute, Shanghai, China. Correspondence to: Yang Yuan <yuanyang@tsinghua.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

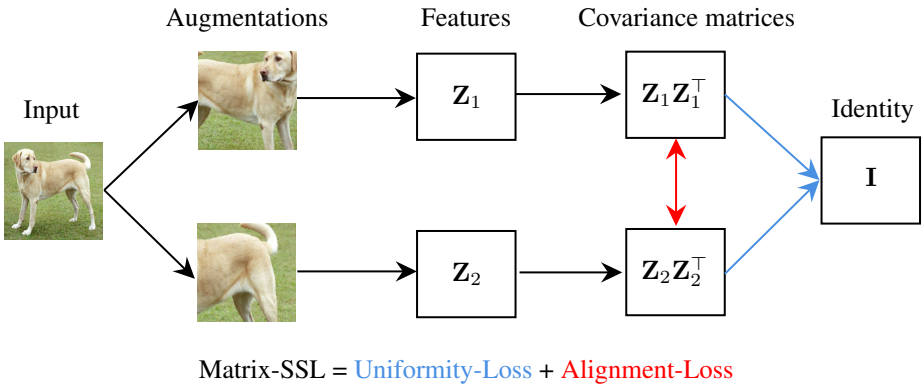


Figure 1. Illustration of the Matrix-SSL architecture. The diagram begins with the image input layer, followed by data augmentations and feature extraction, leading to the formation of covariance matrices ($\mathbf{Z}_1\mathbf{Z}_1^\top$ and $\mathbf{Z}_2\mathbf{Z}_2^\top$).

Could there exist a more encompassing framework that harmonizes the virtues of both contrastive and non-contrastive learning methods?

In this paper, we affirmatively address this question, presenting a method that not only integrates but also enhances the advantages of both contrastive and non-contrastive learning paradigms.

The existing maximum entropy encoding framework, however, does not explicitly differentiate between feature matrices from different branches, hindering its integration with alignment loss. To bridge this gap, we introduce matrix information theory. By extending classical concepts like entropy, Kullback–Leibler (KL) divergence, and cross-entropy to matrix analogs, we offer a richer representation of associated loss functions. Notably, we find that methods like SimSiam, BYOL, Barlow Twins, and MEC can be reinterpreted as utilizing matrix cross-entropy (MCE)-based loss functions, a connection previously unexplored (see Theorem 4.1).

Our proposed algorithm, Matrix-SSL, incorporates matrix alignment loss into non-contrastive methods, leading to improvements in empirical performance. This dual focus provides additional information and a richer signal for representation learning. Matrix-SSL includes Matrix-Uniformity and Matrix-Alignment loss components. Matrix-Uniformity aligns the cross-covariance matrix of feature matrices $\mathbf{Z}_1$ and $\mathbf{Z}_2$ with the identity matrix $\mathbf{I}_d$, while Matrix-Alignment focuses on aligning their auto-covariance matrices (see Figure 1). As a by-product, we observe the closed-form relationship between effective rank and matrix KL, which indicates that effective rank can be a powerful metric for measuring performance for various machine learning methods (see Section 3.4).

In experimental evaluations, our method Matrix-SSL outperforms state-of-the-art methods (SimCLR, BYOL, SimSiam, Barlow Twins, VICReg, etc.) on ImageNet datasets, especially under linear evaluation settings, our method uses only 100 epochs pre-training can outperform SimCLR 100 epochs pre-training by 4.6%. For transfer learning tasks such as COCO detection and COCO instance segmentation, our method outperforms previous SOTA methods such as MoCo v2 and BYOL up to 3% with only 400 epochs compared to 800 epochs pre-training.

We further introduce representation learning into the language modeling regime and use the matrix cross-entropy loss to fine-tune large language models, achieving SOTA results on the GSM8K dataset for mathematical reasoning with a margin of 3.1% over standard cross-entropy loss.

In summary, our contributions can be listed as three-fold:

- We prove the equivalence of MEC loss and matrix uniformity loss (up to constant terms and factors) in non-contrastive learning, and the closed-form relationship between effective rank and matrix KL.
- We provide a unified perspective of uniformity loss plus alignment loss for both contrastive and non-contrastive learning methods.
- We empirically verify our method under various tasks including linear evaluation on image classification tasks, transfer learning on object detection and instance segmentation tasks, and large language model fine-tuning for mathematical reasoning tasks.

2. Related Work

Contrastive and non-contrastive SSL approaches. Contrastive and non-contrastive self-supervised learning methods learn representations based on diverse views or augmentations of inputs, without using human-annotated labels (Chen et al., 2020a; Hjelm et al., 2018; Wu et al., 2018; Tian et al., 2019; Chen & He, 2021; Gao et al., 2021; Bach-

man et al., 2019; Oord et al., 2018; Ye et al., 2019; Henaff, 2020; Misra & Maaten, 2020; Caron et al., 2020; HaoChen et al., 2021; Caron et al., 2021; Li et al., 2021; Zbontar et al., 2021; Tsai et al., 2021b; Bardes et al., 2021; Tian et al., 2020b; Robinson et al., 2021). Such representations can be used for various downstream tasks with remarkable performance.

Theoretical understanding of self-supervised learning. The empirical success of contrastive learning has triggered a surge of theoretical explorations into the contrastive loss (Arora et al., 2019; HaoChen et al., 2021; 2022; Tosh et al., 2020; 2021; Lee et al., 2020; Wang et al., 2022; Nozawa & Sato, 2021; Huang et al., 2021; Tian, 2022; Hu et al., 2022; Tan et al., 2023b). Wang & Isola (2020) shed light on the optimal solutions of the InfoNCE loss, decomposing it as alignment term and uniformity term, contributing to a deeper comprehension of self-supervised learning. In HaoChen et al. (2021); Shen et al. (2022); Wang et al. (2022); Tan et al. (2023b), self-supervised learning methods are examined from a spectral graph perspective. Zimmermann et al. (2021) provides a compelling probabilistic view of contrastive learning, suggesting that it can be seen as an inversion of the data-generating process, which assumes that the ground-truth marginal distribution of the latents of the generative process is uniform.

Various theoretical studies have also investigated non-contrastive methods for self-supervised learning (Wen & Li, 2022; Tian et al., 2021; Garrido et al., 2022; Balestriero & LeCun, 2022; Tsai et al., 2021b; Pokle et al., 2022; Tao et al., 2022; Lee et al., 2021). Garrido et al. (2022) establishes the duality between contrastive and non-contrastive methods. Balestriero & LeCun (2022) reveal the connections between variants of SimCLR, Barlow Twins, and VICReg to ISOMAP, Canonical Correlation Analysis, and Laplacian Eigenmaps, respectively.

Tan et al. (2023a) also use matrix information theory to analyze non-contrastive methods, but they focus on applying α-order mutual information to characterize the loss functions of Barlow Twins and spectral contrastive learning, and extend the analysis to MAE. By contrast, our paper focuses on incorporating alignment loss into the maximum entropy encoding framework.

Neural collapse and dimensional collapse. Papyan et al. (2020) describe the intriguing phenomenon of Neural Collapse (NC), which manifests when training a classification network with cross-entropy loss. This phenomenon can be summarized that all the features of a single class converge to the mean of these features. In addition, the class-means form a simplex equiangular tight frame (ETF). Zhuo et al. (2023) advocate for a comprehensive theoretical understanding of non-contrastive learning through the mechanism of rank differential.

3. Background

Self-supervised learning (SSL) aims to learn meaningful representations from unlabeled data $\{x_i\}_{i=1}^n$, which can be used to enhance performance in various downstream tasks. Prominent SSL methods (architectures) like SimCLR, SimSiam, BYOL, Barlow Twins, and VICReg, employ 2-view augmentations: an online network $\boldsymbol{f}_\theta$ and a target network $\boldsymbol{f}_\phi$. Given a mini-batch $\{\mathbf{x}_i\}_{i=1}^B$, each data point $\mathbf{x}_i$ is augmented using a random transformation $\mathcal{T}$ from a predefined set τ to obtain $\mathbf{x}_i' = \mathcal{T}(\mathbf{x}_i)$. These original and augmented data points are processed through the respective networks to generate feature representations $\mathbf{z}_1^i$ and $\mathbf{z}_2^i$, both residing in $\mathbb{R}^d$. The resulting matrices $\mathbf{Z}_1$ and $\mathbf{Z}_2 \in \mathbb{R}^{d\times B}$ form the basis for the training loss $\mathcal{L}(\mathbf{Z}_1, \mathbf{Z}_2)$, which varies based on the learning paradigm—contrastive or non-contrastive.

3.1. Contrastive Learning

The idea of contrastive learning is to make the representation of similar objects align and dissimilar objects apart. One of the widely adopted losses in contrastive learning is the InfoNCE (Oord et al., 2018) loss, where we use cosine similarity $\operatorname{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v}/(\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2)$:

$$\mathcal{L}_{\text{Contrastive}}(\mathbf{Z}_1, \mathbf{Z}_2) = \sum_i -\log \frac{\exp\left(\operatorname{sim}\left(\mathbf{Z}_1^i, \mathbf{Z}_2^i\right)/\tau\right)}{\sum_{(p,k)\neq(1,i)} \exp\left(\operatorname{sim}\left(\mathbf{Z}_1^i, \mathbf{Z}_p^k\right)/\tau\right)}.$$

Wang & Isola (2020) showed that when the sample size B goes to infinity, $\mathcal{L}_{\text{Contrastive}}$ can be decomposed into two parts. The first part is minimized if and only if $\mathbf{Z}$ is perfectly aligned (alignment loss), while if perfectly uniform encoders exist, they form the exact minimizers of the second part (uniformity loss).

3.2. Non-contrastive Learning

Given a matrix $\mathbf{Z}$, the total coding rate (TCR) (Cover, 1999; Ma et al., 2007) is given by:

$$\mathcal{L}_{\text{TCR}}(\mathbf{Z}) = \frac{1}{2}\log\det\left(\mathbf{I}_d + \frac{d}{B\epsilon^2}\mathbf{Z}\mathbf{Z}^\top\right). \quad (1)$$

Here $(d+B)\mathcal{L}_{\text{TCR}}(\mathbf{Z})$ captures the minimal number of bits for encoding $\mathbf{Z}$ up to ϵ distortion (Cover, 1999; Ma et al., 2007).

For the non-contrastive learning setting, we hope to maximize the total coding rate for the feature embeddings. Given that both the online and target networks are approximations of the feature map $\boldsymbol{f}$, we can use the cross-covariance between $\mathbf{Z}_1$ and $\mathbf{Z}_2$ to approximate $\mathbf{Z}\mathbf{Z}^\top$, resulting in the

maximal entropy coding (MEC) loss (Liu et al., 2022):

$$\begin{aligned}\mathcal{L}_{\text{MEC}} &= -\mu \log\det\left(\mathbf{I}_d + \frac{d}{B\epsilon^2}\mathbf{Z}_1\mathbf{Z}_2^\top\right) \\ &= -\mu \operatorname{tr}\left(\log\left(\mathbf{I}_d + \frac{d}{B\epsilon^2}\mathbf{Z}_1\mathbf{Z}_2^\top\right)\right).\end{aligned} \tag{2}$$

As discussed in (Liu et al., 2022), MEC loss is a natural and general loss that subsumes many non-contrastive learning methods, including SimSiam (Gao et al., 2021), BYOL (Grill et al., 2020), Barlow Twins (Zbontar et al., 2021), and VICReg (Bardes et al., 2021).

3.3. Matrix Information-Theoretic Quantities

Unlike Shannon entropy for random variables, the definition of matrix entropy is not necessarily unique. Specifically, within the domain of quantum information theory, matrix entropy is typically confined to positive semi-definite Hermitian matrices that possess a unit trace. However, our paper aims to extend this definition by incorporating positive semi-definite matrices that are not constrained by unit trace prerequisites, because the matrices may have various traces during optimization.

Definition 3.1 (Matrix entropy for positive semi-definite matrices)**.** For a positive semi-definite matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, the matrix entropy is defined as:

$$\begin{aligned}\mathrm{ME}(\mathbf{A}) &= -\operatorname{tr}(\mathbf{A}\log\mathbf{A}) + \operatorname{tr}(\mathbf{A}) \\ &= -\sum_i \lambda_i \log\lambda_i + \sum_i \lambda_i.\end{aligned}$$

where $\log$ denotes the principal matrix logarithm (Higham, 2008), and λ_i denote the eigenvalues of matrix $\mathbf{A}$. For zero eigenvalues, we define $\log(0) := 0$. Our proposed matrix entropy generalizes the definition of von Neumann entropy (von Neumann, 1932; Witten, 2020), which is restricted to positive semi-definite matrices with unit trace.

Definition 3.2 (Matrix KL divergence for positive semi-definite matrices (Amari, 2014))**.** For two positive semi-definite matrices $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{n\times n}$, the matrix KL divergence is defined as:

$$\mathrm{MKL}(\mathbf{P}||\mathbf{Q}) = \operatorname{tr}(\mathbf{P}\log\mathbf{P} - \mathbf{P}\log\mathbf{Q} - \mathbf{P} + \mathbf{Q}). \tag{3}$$

This definition of matrix KL divergence generalizes the definition of quantum (von Neumann) KL divergence (relative entropy) (von Neumann, 1932; Witten, 2020; Bach, 2022).

Similar to classical cross-entropy based on Shannon information theory, we introduce the matrix cross-entropy as below:

Definition 3.3 (Matrix Cross-Entropy (MCE) for positive semi-definite matrices)**.** For two positive semi-definite matrices $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{n\times n}$, the matrix cross-entropy is defined as:

$$\begin{aligned}\mathrm{MCE}(\mathbf{P},\mathbf{Q}) &= \mathrm{MKL}(\mathbf{P}||\mathbf{Q}) + \mathrm{ME}(\mathbf{P}) \\ &= \operatorname{tr}(-\mathbf{P}\log\mathbf{Q} + \mathbf{Q}).\end{aligned} \tag{4}$$

Lemma 3.4. *For any non-zero matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{A}\mathbf{A}^\top$ is positive semi-definite.*

If not specified, we present proofs in the Appendix A. We employ matrix KL divergence and matrix cross-entropy (MCE) as canonical metrics for comparing positive semi-definite matrices since they have strong minimization properties, just like the classical KL divergence and cross-entropy in Shannon information theory (MKL and MCE are also asymmetric just like the classical ones).

Proposition 3.5 (Minimization property of matrix KL divergence)**.** *For two positive semi-definite matrices $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{n\times n}$, the matrix $\mathbf{Q}$ that minimizes this divergence when $\mathbf{P}$ is fixed and $\mathbf{Q}$ varies over all positive semi-definite matrices is $\mathbf{P}$ itself, i.e.,*

$$\operatorname{argmin}_{\mathbf{Q}\succ 0} \mathrm{MKL}(\mathbf{P}||\mathbf{Q}) = \mathbf{P}. \tag{5}$$

Proposition 3.6 (Minimization property of matrix cross-entropy)**.** *Let $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{n\times n}$ be positive semi-definite matrices. Then, the matrix $\mathbf{Q}$ that minimizes the matrix cross-entropy $\mathrm{MCE}(\mathbf{P},\mathbf{Q})$ when $\mathbf{P}$ is fixed and $\mathbf{Q}$ varies over all positive semi-definite matrices is $\mathbf{P}$ itself, i.e.,*

$$\operatorname{argmin}_{\mathbf{Q}\succ 0} \mathrm{MCE}(\mathbf{P},\mathbf{Q}) = \mathbf{P}. \tag{6}$$

Illustrative example. Consider a batch size $B = 2$ with two augmentation views. Let the representation matrices be $\mathbf{Z}_1 = [\mathbf{a}_1, \mathbf{b}_1] \in \mathbb{R}^{2\times 2}$ for the first view, and $\mathbf{Z}_2 = [\mathbf{a}_2, \mathbf{b}_2] \in \mathbb{R}^{2\times 2}$ for the second view. Suppose $\mathbf{a}_1 = (1, 0)^\top$ and $\mathbf{a}_2 = (0.8, 0.6)^\top$.

Consider two cases:

1. $\mathbf{b}_1 = (0, 1)^\top$ and $\mathbf{b}_2 = (0.6, 0.8)^\top$.
2. $\mathbf{b}_1 = (0.6, 0.8)^\top$ and $\mathbf{b}_2 = (0, 1)^\top$.

In both cases, the typical alignment loss (e.g., BYOL-type MSE loss, $\|\mathbf{a}_1 - \mathbf{a}_2\|^2 + \|\mathbf{b}_1 - \mathbf{b}_2\|^2$) yields a value of 0.8. However, analyzing the covariance matrices $\mathbf{Z}_1\mathbf{Z}_1^\top$ and $\mathbf{Z}_2\mathbf{Z}_2^\top$ reveals more information:

- For Case 1:

$$\mathbf{Z}_1\mathbf{Z}_1^\top = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \quad \mathbf{Z}_2\mathbf{Z}_2^\top = \begin{bmatrix}1 & 0.96\\ 0.96 & 1\end{bmatrix}$$

  Here $\mathrm{MKL}(\mathbf{Z}_1\mathbf{Z}_1^\top \| \mathbf{Z}_2\mathbf{Z}_2^\top) = 2.55$.

- For Case 2:

$$\mathbf{Z}_1\mathbf{Z}_1^\top = \begin{bmatrix}1.36 & 0.48\\ 0.48 & 0.64\end{bmatrix}, \quad \mathbf{Z}_2\mathbf{Z}_2^\top = \begin{bmatrix}0.64 & 0.48\\ 0.48 & 1.36\end{bmatrix}$$

  Here $\mathrm{MKL}(\mathbf{Z}_1\mathbf{Z}_1^\top \| \mathbf{Z}_2\mathbf{Z}_2^\top) = 0.60$.

Matrix information theory, suitable for handling covariance and Gram matrices, allows us to capture these nuanced differences, enabling a more comprehensive understanding of the data representations. Aligning the matrices $\mathbf{Z}_1\mathbf{Z}_1^\top$ and $\mathbf{Z}_2\mathbf{Z}_2^\top$ is beneficial because it can reveal richer training signals beyond the typical vector alignment loss. Even when the vector alignment loss (e.g., BYOL-type MSE loss) yields the same value, the matrix alignment loss, measured by the matrix KL divergence $\mathrm{MKL}(\mathbf{Z}_1\mathbf{Z}_1^\top||\mathbf{Z}_2\mathbf{Z}_2^\top)$, can vary significantly between different cases. This variation provides additional insights into the structural alignment of the representations, ensuring that the learned features capture more detailed and discriminative information about the underlying data distribution.

3.4. Effective Rank

Roy & Vetterli (2007) introduced the concept of effective rank, which provides a real-valued extension of the classical rank.

Definition 3.7 (Effective rank (Roy & Vetterli, 2007))**.** The effective rank of a non-all-zero $\mathbf{A} \in \mathbb{C}^{n\times n}$, denoted $\mathrm{erank}(\mathbf{A})$, is defined as

$$\mathrm{erank}(\mathbf{A}) \triangleq \exp\left\{\mathrm{H}\left(p_1, p_2, \ldots, p_n\right)\right\}, \tag{7}$$

where $p_i = \frac{\sigma_i}{\sum_{k=1}^n \sigma_k}$, $\{\sigma_i | i = 1, \cdots, n\}$ are the singular values of $\mathbf{A}$, and H is the Shannon entropy defined as $\mathrm{H}\left(p_1, p_2, \ldots, p_n\right) = -\sum_{i=1}^n p_i \log p_i$, with the convention that $0 \log 0 \triangleq 0$.

4. On TCR and Matrix KL Divergence

As we mentioned previously, there are two interesting questions about TCR. First, it is not immediately obvious why it is similar to the uniformity loss in contrastive learning. Secondly, one cannot easily integrate matrix alignment loss to directly align the feature covariance matrices in its formulation. In this section, we try to address both problems by building the connection between TCR and MCE/MKL.

Given a batch of B data points $\{x_i\}_{i=1}^B$, and their l_2 normalized representations $\mathbf{Z} = [\boldsymbol{f}(x_1), \cdots, \boldsymbol{f}(x_B)] \in \mathbb{R}^{d\times B}$. We design the following loss function to pursue uniformity, resulting in the following λ-regularized ($\lambda \geq 0$) Uniformity-MCE loss, which is well-defined due to Lemma 3.4:

$$\mathrm{MCE}\left(\frac{1}{d}\mathbf{I}_d + \lambda\mathbf{I}_d, \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I}_d\right), \tag{8}$$

This MCE-based uniformity loss definition captures the distance of regularized covariance matrix $\mathbf{Z}\mathbf{Z}^\top$ to the regularized (scaled) identity matrix and We intentionally introduce the additional regularizer $\lambda \geq 0$ here, because we can prove the closed-form relationship between TCR and MCE/MKL for specific $\lambda > 0$, as follows.

Theorem 4.1 (Main Theorem)**.** *Given a batch of B data points $\{x_i\}_{i=1}^B$, and their l_2 normalized representations $\mathbf{Z} = [\boldsymbol{f}(x_1), \cdots, \boldsymbol{f}(x_B)] \in \mathbb{R}^{d\times B}$. Assume that $\lambda = \frac{\epsilon^2}{d} > 0$ for ϵ, d in TCR loss (1). Then,*

$$\begin{aligned}&\mathrm{MCE}\left(\frac{1}{d}\mathbf{I}_d + \lambda\mathbf{I}_d, \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I}_d\right)\\ =&(1+d\lambda)\left(-\log\lambda + 1 + 2\mathcal{L}_{TCR}(\mathbf{Z})\right),\end{aligned} \tag{9}$$

$$\begin{aligned}&\mathrm{MKL}\left(\frac{1}{d}\mathbf{I}_d + \lambda\mathbf{I}_d \,\middle|\middle|\, \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I}_d\right)\\ =&(1+d\lambda)\left(\log\frac{1+d\lambda}{\lambda d} + 2\mathcal{L}_{TCR}(\mathbf{Z})\right).\end{aligned} \tag{10}$$

Theorem 4.1 shows a deep connection between TCR and MCE/MKL. Indeed, every TCR loss can be transformed into an MCE/MKL loss of the regularized covariance matrix to the scaled identity matrix (but not vice-versa since MCE/MKL has two operands while TCR has only one, and MCE/MKL can also be used for matrix alignment loss introduced in Section 5.1).

Proof sketch, see the full proof in Appendix A.

Proof. For notational simplicity, let

$$\mathcal{L}_{\mathrm{UMCE}} \triangleq \mathrm{MCE}\left(\frac{1}{d}\mathbf{I}_d + \lambda\mathbf{I}_d, \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I}_d\right) \tag{11}$$

Using the definition of MCE, we get:

$$\begin{aligned}&\mathrm{MCE}\left(\frac{1}{d}\mathbf{I}_d + \lambda\mathbf{I}_d, \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I}_d\right)\\ &= \mathrm{tr}\left(-\left(\frac{1}{d}\mathbf{I}_d + \lambda\mathbf{I}_d\right)\log\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I}_d\right)\right)\\ &\quad + \mathrm{tr}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I}_d\right),\end{aligned}$$

Now, let us divide and multiply by λ of the term $-\log\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I}_d\right)$:

$$-\log\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \frac{\epsilon^2}{d}\mathbf{I}_d\right) = -\log\left(\lambda\left(\frac{1}{\lambda B}\mathbf{Z}\mathbf{Z}^\top + \mathbf{I}_d\right)\right),$$

Upon substitution and simplification, we get:

$$\begin{aligned}\mathcal{L}_{\mathrm{UMCE}} &= -(1+d\lambda)\log\lambda + 2(1+d\lambda)\mathcal{L}_{\mathrm{TCR}} + 1 + d\lambda\\ &= (1+d\lambda)\left(-\log\lambda + 1 + 2\mathcal{L}_{\mathrm{TCR}}\right).\end{aligned}$$

This matches the expression given in the proposition for $\mathcal{L}_{\mathrm{UMCE}}$. The other part of the theorem on $\mathrm{MKL}\left(\frac{1}{d}\mathbf{I}_d + \lambda\mathbf{I}_d \,\middle|\middle|\, \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I}_d\right)$ can be proved similarly. □

From Proposition 3.5, Proposition 3.6, and Theorem 4.1, we have the following theorem.

Theorem 4.2 (Minimization property of TCR loss)**.** *Given a batch of B data points $\{x_i\}_{i=1}^B$, and their l_2-normalized representations $\mathbf{Z} = [\boldsymbol{f}(x_1), \cdots, \boldsymbol{f}(x_B)] \in \mathbb{R}^{d\times B}$, the global and unique minimizer under the constraint $\|\mathbf{z}_i\|_2^2 = 1$, for $i \in \{1, 2, \cdots, B\}$ of TCR loss is $\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top = \frac{1}{d}\mathbf{I}_d$.*

In other words, the covariance matrix that minimizes the TCR loss is the (scaled) identity matrix.

5. Matrix Uniformity and Alignment

Based on the discussions in Section 4, we know that TCR loss can be replaced (up to constant terms and factors) by the MCE loss of the (regularized) covariance matrix to the scaled identity matrix. However, if we directly use the covariance matrix of $\mathbf{Z}$, the optimization process might be sub-optimal, as $\mathbf{Z}$ is not empirically aligned to have zero mean. Fortunately, the next theorem states that even if we center the covariance matrix, it will still be aligned with the scaled identity matrix at the maximal effective rank and unit trace.

Theorem 5.1. *Let $\mathbf{x}$ be a random vector with a distribution supported on the unit hypersphere S^{d-1}. If the centered covariance matrix of $\mathbf{x}$, denoted by $\mathbf{C}(\mathbf{x})$, has the maximal possible effective rank d and a trace of at least one, then the expected value of $\mathbf{x}$ will be zero, and $\mathbf{C}(\mathbf{x})$ will equal $\frac{1}{d}\mathbf{I}_d$.*

To achieve matrix information-theoretic uniformity, we propose the following MCE-based uniformity loss, where $\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_2) = \frac{1}{B}\mathbf{Z}_1\mathbf{H}_B\mathbf{Z}_2^\top$ (where $\mathbf{H}_B = \mathbf{I}_B - \frac{1}{B}\mathbf{1_B}\mathbf{1_B}^\top$) represents the centered sample covariance matrix for simplicity:

$$\mathcal{L}_{\text{Matrix-Uniformity}}(\mathbf{Z}_1, \mathbf{Z}_2) = \text{MCE}\left(\frac{1}{d}\mathbf{I}_d, \mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right). \tag{12}$$

The next lemma states why $\mathbf{H}_B$ is the correct centering matrix to use.

Lemma 5.2. *Let $\mathbf{Z}_1, \mathbf{Z}_2 \in \mathbb{R}^{d\times B}$ where d is the dimensionality of the data and B is the number of samples. The cross-covariance matrix $\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_2)$ can be expressed as:*

$$\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right) = \frac{1}{B}\mathbf{Z}_1\mathbf{H}_B\mathbf{Z}_2^\top,$$

where $\mathbf{H}_B = \mathbf{I}_B - \frac{1}{B}\mathbf{1_B}\mathbf{1_B}^\top$ is the centering matrix.

For ease of optimization, a regularization term $\lambda\mathbf{I}_d$ may be added to this cross-covariance matrix to ensure it is non-singular. This adjustment aligns with TCR and MEC methods, differing mainly in mean normalization. An alternative approach is the auto-covariance uniformity loss $\sum_i \text{MCE}\left(\frac{1}{d}\mathbf{I}_d, \mathbf{C}\left(\mathbf{Z}_i, \mathbf{Z}_i\right)\right)$, which is left for future exploration.

5.1. Matrix-SSL: Uniformity and Alignment

To directly pursue the alignment of representations in self-supervised learning, we propose the following loss function using the first-order alignment loss plus the matrix cross-entropy (MCE) between two covariance matrices:

$$\mathcal{L}_{\text{Matrix-Alignment}}(\mathbf{Z}_1, \mathbf{Z}_2) = -\operatorname{tr}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_2)\right) + \gamma \cdot \text{MCE}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1), \mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right). \tag{13}$$

Discussion. When the stop-gradient technique (Gao et al., 2021) is utilized on the target branch $\mathbf{Z}_1$, optimizing the MCE alignment loss is the same as optimizing the matrix KL divergence, since $\text{MCE}(\mathbf{P}, \mathbf{Q}) = \text{MKL}(\mathbf{P}||\mathbf{Q}) + \text{ME}(\mathbf{P})$. We think this can partially answer the effectiveness of stop-gradient (details can be found in Appendix B).

As we have presented an improved loss for uniformity before, now generalizing Wang & Isola (2020)'s understanding of contrastive learning, we propose the matrix information-theoretic uniformity and alignment framework to improve self-supervised learning:

$$\mathcal{L}_{\text{Matrix-SSL}} = \mathcal{L}_{\text{Matrix-Uniformity}} + \mathcal{L}_{\text{Matrix-Alignment}}. \tag{14}$$

6. Effective Rank and Dimensional Collapse

Zhuo et al. (2023) find an intriguing phenomenon that during the optimization course of self-supervised learning, the effective rank of the (empirical) feature covariance matrix consistently increases. This phenomenon can be analyzed with the following proposition.

Proposition 6.1. *Matrix KL divergence of the covariance matrix to the uniform distribution $\frac{1}{d}\mathbf{I}_d$ has the following equality with connection to effective rank.*

$$\begin{aligned}\operatorname{erank}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) &= \frac{d}{\exp\left(\text{MKL}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top \,||\, \frac{1}{d}\mathbf{I}_d\right)\right)} \\ &= \exp\left(\text{VNE}(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top)\right).\end{aligned} \tag{15}$$

Proposition 6.1 captures the closed-form relationship among effective rank and matrix information-theoretic quantities. Note the batch auto-correlation matrix is a positive semi-definite matrix with all of its diagonal 1. As we have mentioned earlier, many dimension-contrastive losses can be understood from the matrix information-theoretic uniformity viewpoint. As such, during training the matrix KL divergence (MCE) minimizes, thus $\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top$ is anticipated to progressively align more with $\frac{1}{d}\mathbf{I}_d$. By the fact that $\frac{1}{d}\mathbf{I}_d$ achieves the maximal possible (matrix) entropy, the rank-increasing phenomenon (Zhuo et al., 2023) can be well understood. Thus we may treat the effective rank as an exact metric to measure the extent of the dimensional collapse.

Feature representations acquired through a deep neural network employing a cross-entropy (CE) loss optimized

by stochastic gradient descent, are capable of attaining zero loss (Du et al., 2018) with arbitrary label assignments (Zhang et al., 2021). A phenomenon known as neural collapse (NC) (Papyan et al., 2020) is observed when training of the neural network continues beyond zero loss with CE. Based on this, we propose to use effective rank as a unified tool to investigate the difference between supervised, contrastive, and non-contrastive methods, more details can be found in Appendix D.

7. Experiments

7.1. Experimental Setup

Experiment details. In this section, we implement our proposed Matrix-SSL method for self-supervised learning tasks on ImageNet (Deng et al., 2009) dataset[1]. We use precisely the same data augmentation protocols and hyperparameters as previous baselines such as BYOL (Grill et al., 2020), SimSiam (Chen & He, 2021) and MEC (Liu et al., 2022), etc. We augment each image twice to get two different views during each training iteration. Similar to MEC (Liu et al., 2022), we select one branch of the Siamese network as the online network and the other branch as the target network, updating the parameters using the exponential moving average method instead of loss backward. The pseudo-code for Matrix-SSL is shown as Algorithm 1.

Model architectures. We use ResNet50 (He et al., 2015) without the last linear layer as the backbone encoder, whose output feature dimension is 2048. Then we use a three-layer MLP with BN(Batch Normalization) (Ioffe & Szegedy, 2015) and ReLU (Nair & Hinton, 2010) as the projector after the encoder, and the projector maintains the feature dimension to be 2048 through three layers. For the online network, we apply an extra two-layer MLP with BN (Ioffe & Szegedy, 2015) and ReLU (Nair & Hinton, 2010) with hidden dimension 512 and output dimension 2048.

Data augmentations. Our augmentation protocol consists of random cropping, color jittering, color dropping (grayscale), left-right flipping, Gaussian blurring, and polarization.

Optimization and hyperparameters. For pre-training, we use SGD optimizer with 2048 batch size, 10^{-5} weight decay, 0.9 momentum, and 4.0 base learning rate, which is scheduled by cosine decay learning rate scheduler (Loshchilov & Hutter, 2016), to optimize the online network over training process. For the momentum used for the exponential moving average process, it is set to be 0.996 to 1 scheduled by another cosine scheduler. As for linear evaluation, we use LARS optimizer (You et al., 2017) with 4096 batch size, 0.9 momentum, no weight decay, and 0.03 base learning rate scheduled by cosine decay learning rate scheduler, to train the linear layer over 100 epochs, and report the performance of last epoch.

[1]The code is available at https://github.com/yifanzhang-pro/Matrix-SSL.

Algorithm 1: PyTorch-style Pseudo-code for Matrix-SSL

```
# f: encoder network
# B: batch size
# L_Matrix-Uniformity: Matrix-Uniformity loss
# L_Matrix-Alignment: Matrix-Alignment loss
# γ: weight ratio used in alignment loss
for X in loader:
   # augment a batch of B images in X
   X_1, X_2 = aug(X), aug(X)

   # calculate l_2 normalized embeddings
   Z_1, Z_2 = f(X_1), f(X_2)

   # calculate uniformity and alignment loss
   uniformity_loss = L_Matrix-Uniformity(Z_1, Z_2)
   alignment_loss = L_Matrix-Alignment(γ)(Z_1, Z_2)

   # calculate loss
   loss = uniformity_loss + alignment_loss

   # optimization step
   loss.backward()
   optimizer.step()
```

7.2. Evaluation Results

Linear evaluation. We follow the standard linear evaluation protocol (Chen et al., 2020a; Grill et al., 2020; Chen & He, 2021). We freeze the parameters of the backbone encoder and then connect a linear classification layer after it, and train the linear layer in the supervised setting. During training, each image is augmented by random cropping, resizing to 224×224, and random horizontal flipping. At test time, each image is resized to 256×256 and center cropped to 224× 224.

The Linear evaluation of the Top-1 accuracy result when pre-trained with 100, 200, and 400 epochs on ImageNet (Deng et al., 2009) dataset was shown in Table 1. Notice that we use ResNet50 backbone as default for a fair comparison. Matrix-SSL consistently outperforms baselines across various pre-training epochs.

Transfer learning. Following the common protocol of previous works (Chen et al., 2020b; Chen & He, 2021; Liu et al., 2022), we finetune the pre-trained models on MS-

Table 1. **Linear evaluation** results (Top-1 accuracy) on ImageNet dataset with different pre-training epochs using ResNet50 backbone. **Bold** means the best, underline means the second.

Method	Pre-training Epochs		
	100	200	400
SimCLR	66.5	68.3	69.8
MoCo v2	67.4	69.9	71.0
BYOL	66.5	70.6	73.2
SwAV	66.5	69.1	70.7
SimSiam	68.1	70.0	70.8
Barlow Twins	67.3	70.2	71.8
VICReg	68.6	–	–
MEC	70.6	71.9	73.5
Matrix-SSL (Ours)	**71.1**	**72.3**	**73.6**

Table 3. **Transfer learning on instance segmentation tasks.** Employing a similar setup as in the detection tasks, we finetune models pre-trained on ImageNet. **Bold** means the best, underline means the second.

Method	AP^{mask}_{50}	AP^{mask}	AP^{mask}_{75}
SimCLR	54.6	33.3	35.3
MoCo v2	55.8	34.4	36.5
BYOL	54.3	33.2	35.0
SwAV	55.2	33.8	35.9
Barlow Twins	56.0	34.3	36.5
SimSiam	56.0	34.4	36.7
VICReg	-	-	36.7
MEC	56.3	34.7	36.8
Matrix-SSL (ours)	**57.5**	**35.6**	**38.0**

COCO (Lin et al., 2014) object detection and instance segmentation tasks. Table 2 and Table 3 summarize experiment results of baseline models and Matrix-SSL. The experiment showed that Matrix-SSL consistently outperformed the baselines. It is worth mentioning that Matrix-SSL was only pre-trained for 400 epochs, but it already performed better than all the baselines pre-trained for 800 epochs. For a fair comparison, we employ a standard 2-view augmentation for all methods, more augmentation views such as $2+6$ views used in SwAV (Caron et al., 2020), $2+2$ views used in I-VNE+ (Kim et al., 2023), and 200 views used in EMP-SSL (Tong et al., 2023) would lead to superior performance and have been theoretically justified (Allen-Zhu & Li, 2020).

Table 2. **Transfer learning on object detection tasks.** We finetune models pre-trained on ImageNet, with the same experiment settings as SimSiam and MEC for a fair comparison.

Method	AP_{50}	AP	AP_{75}
SimCLR	57.7	37.9	40.9
MoCo v2	58.9	39.3	42.5
BYOL	57.8	37.9	40.9
SwAV	58.6	38.4	41.3
Barlow Twins	59.0	39.2	42.5
SimSiam	59.3	39.2	42.1
VICReg	-	40.0	-
MEC	59.8	39.8	43.2
Matrix-SSL (Ours)	**60.8**	**41.0**	**44.2**

Semi-supervised learning. In semi-supervised learning tasks, we noticed that SwAV (Caron et al., 2020), BarlowTwins (Zbontar et al., 2021), and MEC (Liu et al., 2022) all chose different experiment settings and hyperparameters for this task. For a fair comparison, we directly used the same evaluation protocol as MEC and conducted a comparison with semi-supervised learning following 100-epoch pre-training against MEC, since MEC has the best performance in all the baselines on the semi-supervised task. From Table 4, we found that we achieved a significant improvement over MEC in 1% semi-supervised learning, and we are comparable to MEC in the 10% task.

Table 4. Results on semi-supervised learning tasks.

Method	1% Acc@1	1% Acc@5	10% Acc@1	10% Acc@5
MEC	44.442	71.430	63.918	**86.270**
Matrix-SSL	**45.158**	**71.848**	**63.940**	86.172

7.3. Ablation Studies

Alignment loss ratio. We first investigate the impact of different alignment loss ratios (i.e., the γ in Eqn. 14) on performance. We chose the 100-epoch pre-training task for the ablations, and the results are summarized in Table 5. Interestingly, setting $\gamma = 1$ achieves the best linear evaluation performance, so we set the ratio to 1 as the default.

Table 5. Ablations on linear probing (%) with various γ.

γ	0	0.3	0.5	0.6	1	1.3	1.5
Acc.	70.6	70.7	71.0	70.9	**71.1**	70.8	70.8

Taylor expansion order. We investigat the effect of the Taylor expansion order of matrix logarithm implementation (which is well-defined according to Theorem A.1) on linear evaluation tasks, We keep most of the settings unchanged, except the Taylor expansion order. The results are summarized in Table 6. As shown in the table, we found that Matrix-SSL performs best when the Taylor expansion order is 4 in this setting, so we chose 4 as the default parameter.

Table 6. Results of different Taylor expansion orders for linear evaluation results.

Taylor expansion order	3	4	5
Top-1 accuracy	70.9	**71.1**	71.0

8. Matrix Cross-Entropy for Large Language Models

We further introduce representation learning into the language modeling regime and use the matrix cross-entropy loss to fine-tune large language models by considering how to incorporate the information within feature representations in designing the loss functions.

The main intuition behind our method is that the similarity among the representation vector of different words (tokens) can be utilized to address the **synonym phenomenon** and **polysemous phenomenon** within natural language. For example, "Let **'s** think step by step" should be similar to "Let **us** think step by step". The classical cross-entropy loss hasn't captured this intricate part.

Consider the target distribution $\mathbf{p}$ given by the training corpus (which is typically one-hot) and the output distribution $\mathbf{q}$ given by the output of the language model. Suppose we have l_2 normalized representation vectors $\mathbf{e_i} \in \mathbb{R}^d$ (column vectors) for tokens $v_i, i \in [n]$, where n is the vocabulary size. One could use LM head embeddings, word embeddings, or any other representation vectors of the models. In our experiments, we use the LM head embeddings as default.

For auto-regressive LLMs with tokens $k \in \{1, 2, \cdots, K\}$, we define positive semi-definite matrices $\mathbf{P} \in \mathbb{R}^{d\times d}$ and $\mathbf{Q} \in \mathbb{R}^{d\times d}$ as below:

$$\mathbf{P}^{(k)} = \sum_i \left(p_i^{(k)} \cdot \mathbf{e}_i \mathbf{e}_i^\top\right), \quad \mathbf{Q}^{(k)} = \sum_j \left(q_j^{(k)} \cdot \mathbf{e}_j \mathbf{e}_j^\top\right).$$

Then we define the following loss as our objective (since $\operatorname{tr}\left(\mathbf{Q}^{(k)}\right)$ are constant):

$$\begin{aligned}\mathcal{L}_{\text{Matrix-LLM}} &= \sum_k \operatorname{CE}(\mathbf{p}^{(k)}, \mathbf{q}^{(k)}) + \sum_k \operatorname{MCE}(\mathbf{P}^{(k)}, \mathbf{Q}^{(k)}) \\ &= -\sum_k \sum_i p_i^{(k)} \log q_i^{(k)} - \sum_k \operatorname{tr}(\mathbf{P}^{(k)} \log \mathbf{Q}^{(k)}).\end{aligned} \tag{16}$$

8.1. Experiments on Fine-Tuning LLMs

Training Pipeline. We use Llemma-7B (Azerbayev et al., 2023) as the base model, which is continued pre-trained on the Proof-Pile-2 dataset (Paster et al., 2023) using the CodeLLaMA model (Touvron et al., 2023). We then use $\mathcal{L}_{\text{Matrix-LLM}}$ to fine-tune it on the MetaMath dataset (Yu et al., 2023).

Table 7. Performance comparison of various models on GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) dataset. MM denotes instruction fine-tuned with the MetaMathQA dataset (Yu et al., 2023).

Model	**Meth.**	**GSM8K** (%)	**MATH** (%)
Minerva 8B	CE	16.2	14.1
Minerva 62B	CE	52.4	27.6
Minerva 540B	CE	58.8	**33.6**
WizardMath 7B	RL	54.9	10.7
WizardMath 13B	RL	63.9	14.0
WizardMath 70B	RL	**81.6**	22.7
LLaMA2 70B	CE	56.8	13.5
MetaMath 7B	CE	66.5	19.8
Llemma 7B	CE	36.4	18.0
Llemma-MM 7B	CE	69.2	30.0
Llemma-MM 7B	$\mathcal{L}_{\text{Matrix-LLM}}$	**72.3** (+3.1)	**30.2** (+0.2)

Experimental Results. We evaluated the performance of different models on the mathematical reasoning dataset GSM8K (Cobbe et al., 2021) and MATH dataset (Hendrycks et al., 2021), using different loss functions and training methods. The results are shown in Table 7. We compared our results against baseline methods, including Minerva (Lewkowycz et al., 2022), WizardMath (Luo et al., 2023), and Llemma (Azerbayev et al., 2023) fine-tuned with MetaMath (Yu et al., 2023) dataset using classical cross-entropy (CE).

9. Conclusion

In this paper, we provide a matrix information-theoretic perspective for understanding and improving self-supervised learning methods. We are confident that our perspective will not only offer a refined and alternative comprehension of self-supervised learning methods but will also act as a catalyst for the design of increasingly robust and effective algorithms in the future.

Acknowledgment

Yang Yuan is supported by the Ministry of Science and Technology of the People's Republic of China, the 2030 Innovation Megaprojects "Program on New Generation Artificial Intelligence" (Grant No. 2021AAA0150000). Weiran Huang is supported by the 2023 CCF-Baidu Open Fund and Microsoft Research Asia.

We would also like to express our sincere gratitude to the reviewers of ICML 2024 for their insightful and constructive feedback. Their valuable comments have greatly contributed to improving the quality of our work.

Impact Statement

This paper aims to contribute to the advancement of machine learning by introducing novel approaches to self-supervised

learning. While this work primarily seeks to enrich research within the field, it acknowledges the potential broader societal implications inherent in any advancement in machine learning. However, specific societal consequences are not directly foreseeable at this stage.

References

Allen-Zhu, Z. and Li, Y. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020. 8

Amari, S.-i. Information geometry of positive measures and positive-definite matrices: Decomposable dually flat structure. *Entropy*, 16(4):2131–2145, 2014. 4

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, 2019. 3

Azerbayev, Z., Schoelkopf, H., Paster, K., Santos, M. D., McAleer, S., Jiang, A. Q., Deng, J., Biderman, S., and Welleck, S. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023. 9

Bach, F. Information theory with kernel methods. *IEEE Transactions on Information Theory*, 2022. 4

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019. 1, 2

Balestriero, R. and LeCun, Y. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *arXiv preprint arXiv:2205.11508*, 2022. 3

Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 1, 3, 4

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 3, 8

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021. 3

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607. PMLR, 2020a. 1, 2, 7, 22

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021. 1, 2, 7, 22

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b. 7

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 9

Cover, T. M. *Elements of information theory*. John Wiley & Sons, 1999. 3

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009. 7

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018. 7, 19

Galanti, T., György, A., and Hutter, M. On the role of neural collapse in transfer learning. *arXiv preprint arXiv:2112.15121*, 2021. 19

Gao, T., Yao, X., and Chen, D. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021. 2, 4, 6

Garrido, Q., Chen, Y., Bardes, A., Najman, L., and Lecun, Y. On the duality between contrastive and non-contrastive self-supervised learning. *arXiv preprint arXiv:2206.02574*, 2022. 3

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Iformation Processing Systems*, 33:21271–21284, 2020. 1, 4, 7, 22

Hall, B. C. *Lie groups, Lie algebras, and representations*. Springer, 2013. 18

HaoChen, J. Z., Wei, C., Gaidon, A., and Ma, T. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021. 1, 3

HaoChen, J. Z., Wei, C., Kumar, A., and Ma, T. Beyond separability: Analyzing the linear transferability of contrastive representations to related subpopulations. *Advances in Neural Information Processing Systems*, 2022. 3

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385. 7

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020. 1

Henaff, O. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pp. 4182–4192. PMLR, 2020. 3

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021. 9

Higham, N. J. *Functions of matrices: theory and computation*. SIAM, 2008. 4

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018. 1, 2

Hu, T., Liu, Z., Zhou, F., Wang, W., and Huang, W. Your contrastive learning is secretly doing stochastic neighbor embedding. *arXiv preprint arXiv:2205.14814*, 2022. 3

Hua, T. Simsiam. https://github.com/PatrickHua/SimSiam, 2021. 19, 22

Huang, W., Yi, M., and Zhao, X. Towards the generalization of contrastive self-supervised learning. *arXiv preprint arXiv:2111.00743*, 2021. 3

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167. 7

Kim, J., Kang, S., Hwang, D., Shin, J., and Rhee, W. Vne: An effective method for improving deep representation by manipulating eigenvalue distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3799–3810, 2023. 8

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009. 22

Lee, J. D., Lei, Q., Saunshi, N., and Zhuo, J. Predicting what you already know helps: Provable self-supervised learning. *arXiv preprint arXiv:2008.01064*, 2020. 3

Lee, J. D., Lei, Q., Saunshi, N., and Zhuo, J. Predicting what you already know helps: Provable self-supervised learning. *Advances in Neural Information Processing Systems*, 34:309–323, 2021. 3

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., and Misra, V. Solving quantitative reasoning problems with language models, 2022. 9

Li, Y., Pogodin, R., Sutherland, D. J., and Gretton, A. Self-supervised learning with kernel dependence maximization. *Advances in Neural Information Processing Systems*, 34:15543–15556, 2021. 3

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312. 8

Liu, X., Wang, Z., Li, Y.-L., and Wang, S. Self-supervised learning via maximum entropy coding. *Advances in Neural Information Processing Systems*, 35:34091–34105, 2022. 1, 4, 7, 8

Logeswaran, L. and Lee, H. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*, 2018. 1

Loshchilov, I. and Hutter, F. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016. URL http://arxiv.org/abs/1608.03983. 7

Luo, H., Sun, Q., Xu, C., Zhao, P., Lou, J., Tao, C., Geng, X., Lin, Q., Chen, S., and Zhang, D. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct, 2023. 9

Ma, J., You, C., Reddi, S. J., Jayasumana, S., Jain, H., Yu, F., Chang, S.-F., and Kumar, S. Do we need neural collapse? learning diverse features for fine-grained and long-tail classification. *OpenReviewNet*, 2023. 19

Ma, Y., Derksen, H., Hong, W., and Wright, J. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE transactions on pattern analysis and machine intelligence*, 29(9):1546–1562, 2007. 3

Misra, I. and Maaten, L. v. d. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2020. 3

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 2010. 7

Nozawa, K. and Sato, I. Understanding negative samples in instance discriminative self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34:5784–5797, 2021. 3

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

Papyan, V., Han, X., and Donoho, D. L. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020. 3, 7, 19, 20, 22

Paster, K., Santos, M. D., Azerbayev, Z., and Ba, J. Openwebmath: An open dataset of high-quality mathematical web text. *arXiv preprint arXiv:2310.06786*, 2023. 9

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011. 22

Pokle, A., Tian, J., Li, Y., and Risteski, A. Contrasting the landscape of contrastive and non-contrastive learning. *arXiv preprint arXiv:2203.15702*, 2022. 3

Robinson, J. D., Chuang, C.-Y., Sra, S., and Jegelka, S. Contrastive learning with hard negative samples. In *ICLR*, 2021. 3

Roy, O. and Vetterli, M. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pp. 606–610. IEEE, 2007. 5, 21

Shen, K., Jones, R. M., Kumar, A., Xie, S. M., HaoChen, J. Z., Ma, T., and Liang, P. Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 19847–19878. PMLR, 2022. 3

Tan, Z., Yang, J., Huang, W., Yuan, Y., and Zhang, Y. Information flow in self-supervised learning. *arXiv preprint arXiv:2309.17281*, 2023a. 3

Tan, Z., Zhang, Y., Yang, J., and Yuan, Y. Contrastive learning is spectral clustering on similarity graph. *arXiv preprint arXiv:2303.15103*, 2023b. 1, 3, 20

Tao, C., Wang, H., Zhu, X., Dong, J., Song, S., Huang, G., and Dai, J. Exploring the equivalence of siamese self-supervised learning via a unified gradient framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14431–14440, 2022. 3

Tian, Y. Deep contrastive learning is provably (almost) principal component analysis. *arXiv preprint arXiv:2201.12680*, 2022. 3

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 2

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 776–794. Springer, 2020a. 1

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020b. 3

Tian, Y., Chen, X., and Ganguli, S. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pp. 10268–10278. PMLR, 2021. 3

Tong, S., Chen, Y., Ma, Y., and Lecun, Y. Emp-ssl: Towards self-supervised learning in one training epoch. *arXiv preprint arXiv:2304.03977*, 2023. 8

Tosh, C., Krishnamurthy, A., and Hsu, D. Contrastive estimation reveals topic posterior information to linear models. *arXiv:2003.02234*, 2020. 3

Tosh, C., Krishnamurthy, A., and Hsu, D. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pp. 1179–1206. PMLR, 2021. 3

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 9

Tsai, Y. H., Bai, S., Morency, L., and Salakhutdinov, R. A note on connecting barlow twins with negative-sample-free contrastive learning. *CoRR*, abs/2104.13712, 2021a. URL https://arxiv.org/abs/2104.13712. 22

Tsai, Y.-H. H., Bai, S., Morency, L.-P., and Salakhutdinov, R. A note on connecting barlow twins with negative-sample-free contrastive learning. *arXiv preprint arXiv:2104.13712*, 2021b. 3

van der Maaten, L. and Hinton, G. E. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9: 2579–2605, 2008. 22

von Neumann, J. Mathematische grundlagen der quantenmechanik, 1932. 4

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020. 1, 3, 6

Wang, Y., Zhang, Q., Wang, Y., Yang, J., and Lin, Z. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. *arXiv preprint arXiv:2203.13457*, 2022. 3

Wen, Z. and Li, Y. The mechanism of prediction head in non-contrastive self-supervised learning. *arXiv preprint arXiv:2205.06226*, 2022. 3

Witten, E. A mini-introduction to information theory. *La Rivista del Nuovo Cimento*, 43(4):187–227, 2020. 4

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018. 1, 2

Ye, M., Zhang, X., Yuen, P. C., and Chang, S.-F. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6210–6219, 2019. 3

You, Y., Gitman, I., and Ginsburg, B. Scaling SGD batch size to 32k for imagenet training. *CoRR*, abs/1708.03888, 2017. URL http://arxiv.org/abs/1708.03888. 7

Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., Kwok, J. T., Li, Z., Weller, A., and Liu, W. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023. 9

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021. 1, 3, 4, 8, 22

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 7, 19

Zhou, J., You, C., Li, X., Liu, K., Liu, S., Qu, Q., and Zhu, Z. Are all losses created equal: A neural collapse perspective. *arXiv preprint arXiv:2210.02192*, 2022. 20

Zhu, Z., Ding, T., Zhou, J., Li, X., You, C., Sulam, J., and Qu, Q. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34:29820–29834, 2021. 19, 20

Zhuo, Z., Wang, Y., Ma, J., and Wang, Y. Towards a unified theoretical understanding of non-contrastive learning via rank differential mechanism. In *The Eleventh International Conference on Learning Representations*, 2023. 3, 6, 22

Zimmermann, R. S., Sharma, Y., Schneider, S., Bethge, M., and Brendel, W. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pp. 12979–12990. PMLR, 2021. 3

A. Appendix for Proofs

Proof of Lemma 3.4.

Proof. Consider any non-zero matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. We want to show that $\mathbf{A}\mathbf{A}^\top$ is positive semi-definite.

Recall that a matrix $\mathbf{B}$ is positive semi-definite if for all vectors $\mathbf{x} \in \mathbb{R}^m$, it holds that $\mathbf{x}^\top \mathbf{B}\mathbf{x} \geq 0$. We will apply this definition to $\mathbf{A}\mathbf{A}^\top$.

Consider any vector $\mathbf{x} \in \mathbb{R}^m$. We compute $\mathbf{x}^\top(\mathbf{A}\mathbf{A}^\top)\mathbf{x}$ as follows:

$$\begin{aligned}\mathbf{x}^\top(\mathbf{A}\mathbf{A}^\top)\mathbf{x} &= (\mathbf{x}^\top\mathbf{A})(\mathbf{A}^\top\mathbf{x}) \\ &= \|\mathbf{A}^\top\mathbf{x}\|^2.\end{aligned}$$

The last equality holds because the expression $(\mathbf{x}^\top\mathbf{A})(\mathbf{A}^\top\mathbf{x})$ represents the squared norm of the vector $\mathbf{A}^\top\mathbf{x}$.

Since the squared norm of any vector is always non-negative, $\|\mathbf{A}^\top\mathbf{x}\|^2 \geq 0$ for any $\mathbf{x} \in \mathbb{R}^m$.

Therefore, $\mathbf{x}^\top(\mathbf{A}\mathbf{A}^\top)\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^m$, which means that $\mathbf{A}\mathbf{A}^\top$ is positive semi-definite.

This completes the proof. □

Proof of Proposition 3.5.

Proof. We consider the matrix KL divergence $\mathrm{MKL}(\mathbf{P}||\mathbf{Q})$ for positive semi-definite matrices $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{n \times n}$. Our goal is to show that this function attains its minimum when $\mathbf{Q} = \mathbf{P}$.

First, we calculate the gradient of $\mathrm{MKL}(\mathbf{P}||\mathbf{Q})$ with respect to $\mathbf{Q}$. Utilizing the properties of the matrix logarithm and trace, we find

$$\nabla_{\mathbf{Q}} \mathrm{MKL}(\mathbf{P}||\mathbf{Q}) = -\mathbf{P}\mathbf{Q}^{-1} + \mathbf{I},$$

where $\mathbf{I}$ is the identity matrix.

Setting this gradient to zero, we obtain the condition for stationary points:

$$-\mathbf{P}\mathbf{Q}^{-1} + \mathbf{I} = \mathbf{0} \implies \mathbf{P}\mathbf{Q}^{-1} = \mathbf{I}.$$

Multiplying both sides of this equation by $\mathbf{Q}$ yields $\mathbf{Q} = \mathbf{P}$, indicating that $\mathbf{Q} = \mathbf{P}$ is a stationary point of the function.

To confirm that $\mathbf{Q} = \mathbf{P}$ is indeed a minimum, we examine the second-order conditions. The Hessian of $\mathrm{MKL}(\mathbf{P}||\mathbf{Q})$, computed as

$$\nabla^2_{\mathbf{Q}} \mathrm{MKL}(\mathbf{P}||\mathbf{Q}) = \mathbf{P}\mathbf{Q}^{-2},$$

is positive semi-definite. This is because for any non-zero matrix $\mathbf{X} \in \mathbb{R}^{n \times n}$, the expression

$$\mathbf{X}^\top(\mathbf{P}\mathbf{Q}^{-2})\mathbf{X}$$

is non-negative, given that both $\mathbf{P}$ and $\mathbf{Q}^{-2}$ are positive semi-definite. Therefore, $\mathrm{MKL}(\mathbf{P}||\mathbf{Q})$ is convex in $\mathbf{Q}$.

Given the convexity of the function and the identification of a stationary point at $\mathbf{Q} = \mathbf{P}$, we can conclude that this point is indeed the global minimum of the function over the domain of positive semi-definite matrices.

Hence, we conclude that

$$\mathrm{argmin}_{\mathbf{Q} \succ 0} \mathrm{MKL}(\mathbf{P}||\mathbf{Q}) = \mathbf{P},$$

thereby completing the proof. □

Proof of Proposition 3.6.

Proof. The matrix cross-entropy between two positive semi-definite matrices $\mathbf{P}$ and $\mathbf{Q}$ is defined as:

$$\mathrm{MCE}(\mathbf{P}, \mathbf{Q}) = \mathrm{tr}(-\mathbf{P}\log\mathbf{Q} + \mathbf{Q}).$$

To find the matrix $\mathbf{Q}$ that minimizes $\mathrm{MCE}(\mathbf{P}, \mathbf{Q})$, we compute the derivative of MCE with respect to $\mathbf{Q}$. The derivative of the matrix cross-entropy is given by:

$$\frac{\partial \, \mathrm{MCE}}{\partial \mathbf{Q}} = -\mathbf{P}\mathbf{Q}^{-1} + \mathbf{I},$$

where we utilized the matrix calculus result that the derivative of $\log \mathbf{Q}$ with respect to $\mathbf{Q}$ is $\mathbf{Q}^{-1}$.

Setting this derivative to zero for optimality, we get:

$$-\mathbf{P}\mathbf{Q}^{-1} + \mathbf{I} = \mathbf{0} \Longrightarrow \mathbf{P}\mathbf{Q}^{-1} = \mathbf{I}.$$

Multiplying both sides by $\mathbf{Q}$, we obtain:

$$\mathbf{P} = \mathbf{Q}.$$

To confirm that $\mathbf{Q} = \mathbf{P}$ is indeed a minimum, we examine the second-order conditions, the proof is similar to Proof A for Proposition 3.5. Therefore, we conclude that the matrix $\mathbf{Q}$ minimizing the matrix cross-entropy $\mathrm{MCE}(\mathbf{P}, \mathbf{Q})$ is $\mathbf{P}$ itself, i.e.,

$$\mathrm{argmin}_{\mathbf{Q} \succ 0} \, \mathrm{MCE}(\mathbf{P}, \mathbf{Q}) = \mathbf{P}.$$

This completes the proof. □

Proof of Theorem 4.1.

Proof. First, begin with $\mathcal{L}_{\text{UMCE}}$:

$$\mathcal{L}_{\text{UMCE}} = \mathrm{MCE}\left(\frac{1}{d}\mathbf{I}_d + \lambda \mathbf{I}_d, \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda \mathbf{I}_d\right),$$

Using the definition of MCE, we get:

$$\mathcal{L}_{\text{UMCE}} = \mathrm{tr}\left(-\left(\frac{1}{d}\mathbf{I}_d + \lambda \mathbf{I}_d\right) \log\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda \mathbf{I}_d\right) + \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda \mathbf{I}_d\right),$$

Now, let us divide and multiply by λ of the term $-\log\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda \mathbf{I}_d\right)$:

$$-\log\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \frac{\epsilon^2}{d}\mathbf{I}_d\right) = -\log\left(\lambda\left(\frac{1}{\lambda B}\mathbf{Z}\mathbf{Z}^\top + \mathbf{I}_d\right)\right),$$

Now, factor out λ:

$$-\log\left(\lambda\left(\frac{1}{\lambda B}\mathbf{Z}\mathbf{Z}^\top + \mathbf{I}_d\right)\right) = -\log(\lambda)\mathbf{I}_d - \log\left(\frac{1}{\lambda B}\mathbf{Z}\mathbf{Z}^\top + \mathbf{I}_d\right),$$

Since $\mathcal{L}_{\text{TCR}} = \frac{1}{2}\log\det\left(\mathbf{I}_d + \frac{d}{B\epsilon^2}\mathbf{Z}\mathbf{Z}^\top\right)$, we can rewrite this term in the form of $\mathcal{L}_{\text{TCR}}$.

$$\mathrm{tr}\left(-\log\left(\frac{1}{\lambda B}\mathbf{Z}\mathbf{Z}^\top + \mathbf{I}_d\right)\right) = \mathrm{tr}\left(-\log\left(\mathbf{I}_d + \frac{d}{B\epsilon^2}\mathbf{Z}\mathbf{Z}^\top\right)\right) = 2\mathcal{L}_{\text{TCR}},$$

Upon substitution, it becomes:

$$\mathcal{L}_{\text{UMCE}} = -\mathrm{tr}\left(\left(\frac{1}{d}\mathbf{I}_d + \lambda \mathbf{I}_d\right)(\log(\lambda)\mathbf{I}_d)\right) + 2(1 + d\lambda)\mathcal{L}_{\text{TCR}} + \mathrm{tr}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda \mathbf{I}_d\right),$$

Simplifying, we get:

$$\begin{aligned}\mathcal{L}_{\text{UMCE}} &= -(1 + d\lambda)\log\lambda + 2(1 + d\lambda)\mathcal{L}_{\text{TCR}} + 1 + d\lambda \\ &= (1 + d\lambda)\left(-\log\lambda + 1 + 2\mathcal{L}_{\text{TCR}}\right).\end{aligned}$$

This matches the expression given in the proposition for $\mathcal{L}_{\text{UMCE}}$.

For $\mathcal{L}_{\text{UMKL}}$, Using the definition of Matrix KL divergence, we have:

$$\begin{aligned}\mathcal{L}_{\text{UMKL}} &= \text{MKL}\left(\frac{1}{d}\mathbf{I}_d + \lambda\mathbf{I}_d \,\middle|\middle|\, \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I}_d\right), \\ &= \text{MCE}\left(\frac{1}{d}\mathbf{I}_d + \lambda\mathbf{I}_d, \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I}_d\right) + \text{tr}\left(\mathbf{P}\log\mathbf{P} - \mathbf{P}\right),\end{aligned}$$

where $\mathbf{P}$ denotes $\frac{1}{d}\mathbf{I}_d + \lambda\mathbf{I}_d$.

Now, we simplify $\text{tr}\left(\mathbf{P}\log\mathbf{P} - \mathbf{P}\right)$. We know that $\mathbf{P} = \frac{1}{d}\mathbf{I}_d + \lambda\mathbf{I}_d = \left(\frac{1}{d} + \lambda\right)\mathbf{I}_d$.

Since $\mathbf{P}$ is a diagonal matrix with all diagonal entries being $\frac{1}{d} + \lambda$, its matrix logarithm $\log\mathbf{P}$ will also be a diagonal matrix with all diagonal entries being $\log\left(\frac{1}{d} + \lambda\right)$.

Thus, $\text{tr}\left(\mathbf{P}\log\mathbf{P} - \mathbf{P}\right)$ can be simplified as follows:

$$\text{tr}\left(\mathbf{P}\log\mathbf{P} - \mathbf{P}\right) = \text{tr}\left(\left(\frac{1}{d} + \lambda\right)\mathbf{I}_d\left(\log\left(\frac{1}{d} + \lambda\right)\mathbf{I}_d\right) - \left(\frac{1}{d} + \lambda\right)\mathbf{I}_d\right),$$

Since the diagonal matrix $\mathbf{I}_d$ has d ones along its diagonal, the trace operation essentially multiplies each term by d. Therefore, we can write:

$$\text{tr}(\mathbf{P}\log\mathbf{P} - \mathbf{P}) = d\left(\left(\frac{1}{d} + \lambda\right)\log\left(\frac{1}{d} + \lambda\right) - \left(\frac{1}{d} + \lambda\right)\right),$$

Further simplifying, we get:

$$\begin{aligned}\text{tr}(\mathbf{P}\log\mathbf{P} - \mathbf{P}) &= d\left(\frac{1}{d} + \lambda\right)\log\left(\frac{1}{d} + \lambda\right) - d\left(\frac{1}{d} + \lambda\right) \\ &= (1 + d\lambda)(\log(1 + d\lambda) - \log d - 1),\end{aligned}$$

Now, we can rewrite $\mathcal{L}_{\text{UMKL}}$ using this result:

$$\begin{aligned}\mathcal{L}_{\text{UMKL}} &= \mathcal{L}_{\text{UMCE}} + \text{tr}\left(\mathbf{P}\log\mathbf{P} - \mathbf{P}\right) \\ &= \mathcal{L}_{\text{UMCE}} + (1 + d\lambda)(\log(1 + d\lambda) - \log d - 1) \\ &= -(1 + d\lambda)\log\lambda + 2(1 + d\lambda)\mathcal{L}_{\text{TCR}} + 1 + d\lambda + (1 + d\lambda)(\log(1 + d\lambda) - \log d - 1) \\ &= -(1 + d\lambda)\log\lambda + 2(1 + d\lambda)\mathcal{L}_{\text{TCR}} + (1 + d\lambda)\log(1 + d\lambda) - (1 + d\lambda)\log d \\ &= (1 + d\lambda)(-\log\lambda + 2\mathcal{L}_{\text{TCR}} + \log(1 + d\lambda) - \log d) \\ &= (1 + d\lambda)(\log\frac{1 + d\lambda}{\lambda d} + 2\mathcal{L}_{\text{TCR}}).\end{aligned}$$

This equation represents $\mathcal{L}_{\text{UMKL}}$ in terms of $\mathcal{L}_{\text{TCR}}$ and other constants d, λ, and B, thus fulfilling the proposition. □

Proof of Theorem 4.2.

Proof. Here we present an alternative proof without resorting to other literature. To prove the theorem, we examine the form of the TCR loss:

$$\mathcal{L}_{\text{TCR}} = -\frac{1}{2}\log\det\left(\mathbf{I}_d + \frac{d}{B\epsilon^2}\mathbf{Z}\mathbf{Z}^\top\right),$$

where $\mathbf{Z} = [\boldsymbol{f}(x_1), \cdots, \boldsymbol{f}(x_B)] \in \mathbb{R}^{d\times B}$.

We note that $\mathbf{Z}\mathbf{Z}^\top$ is a positive semi-definite matrix, as it is the product of a matrix and its transpose. Hence, all its eigenvalues are non-negative. Let these eigenvalues be denoted by $\lambda_1, \lambda_2, \ldots, \lambda_d$.

The determinant of $\mathbf{I}_d + \frac{d}{B\epsilon^2}\mathbf{Z}\mathbf{Z}^\top$ can then be expressed as the product of its eigenvalues:

$$\det\left(\mathbf{I}_d + \frac{d}{B\epsilon^2}\mathbf{Z}\mathbf{Z}^\top\right) = \prod_{i=1}^{d}(1 + \frac{d}{B\epsilon^2}\lambda_i).$$

Since logarithm is a monotonically increasing function, minimizing $\mathcal{L}_{\text{TCR}}$ is equivalent to maximizing the product of $(1 + \frac{d}{B\epsilon^2}\lambda_i)$ terms.

Applying the arithmetic mean-geometric mean inequality, we find that the product of the eigenvalues (and thus the determinant) is maximized when all eigenvalues are equal, i.e., $\lambda_i = \frac{B}{d}$ for all i. Therefore, the matrix that maximizes this determinant under the given constraints is one where all eigenvalues are $\frac{B}{d}$.

Hence, the global and unique minimizer of the TCR loss under the constraint $\|\mathbf{z}_i\|_2^2 = 1$ is achieved when $\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top$ has eigenvalues equal to $\frac{1}{d}$, which corresponds to $\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top = \frac{1}{d}\mathbf{I}_d$. □

Proof of Theorem 5.1.

Proof. Based on the definition of effective rank presented in Section 3.4, a maximal effective rank of d implies that the covariance matrix has d non-negligible eigenvalues.

Let $\mathbf{x} = [x_1, x_2, \ldots, x_d]^\top$ be a random vector on S^{d-1}. The covariance matrix $\mathbf{C}(\mathbf{x})$ of $\mathbf{x}$ is defined as $\mathbb{E}[\mathbf{x}\mathbf{x}^\top] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}]^\top$.

The trace of $\mathbf{C}(\mathbf{x})$, which is the sum of its eigenvalues, must be at least 1. Given the maximal effective rank d, each of these d eigenvalues must be equal (denote this common value as λ), resulting in $\mathbf{C}(\mathbf{x}) = \lambda\mathbf{I}_d$.

From above, we find that $\mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \lambda\mathbf{I}_d$. Noticing that $\text{tr}(\mathbf{C}(\mathbf{x})) = 1 - \|\mathbb{E}[\mathbf{x}]\|^2 \leq 1$ and the trace at least 1 assumption, the trace of this matrix, which is $d\lambda$, must be equal to 1, implying $\lambda = \frac{1}{d}$.

Thus, we conclude that if the covariance matrix of $\mathbf{x}$ has the maximal possible effective rank of d and its trace is at least one, then the expected value of $\mathbf{x}$ is zero, and the covariance matrix $\mathbf{C}(\mathbf{x})$ is $\frac{1}{d}\mathbf{I}_d$. □

Proof of Lemma 5.2.

Proof. To prove the lemma, we first apply the centering matrix $\mathbf{H}_B$ to $\mathbf{Z}_1$ and $\mathbf{Z}_2$ as follows:

$$\bar{\mathbf{Z}}_1 = \mathbf{Z}_1\mathbf{H}_B,$$
$$\bar{\mathbf{Z}}_2 = \mathbf{Z}_2\mathbf{H}_B.$$

These equations remove the mean of each row, effectively centering the data.

The cross-covariance matrix for the centered data $\bar{\mathbf{Z}}_1$ and $\bar{\mathbf{Z}}_2$ is then given by:

$$\mathbf{C}(\bar{\mathbf{Z}}_1, \bar{\mathbf{Z}}_2) = \frac{1}{B}\bar{\mathbf{Z}}_1\bar{\mathbf{Z}}_2^\top.$$

Substituting the expressions for $\bar{\mathbf{Z}}_1$ and $\bar{\mathbf{Z}}_2$, we get:

$$\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right) = \frac{1}{B}(\mathbf{Z}_1\mathbf{H}_B)(\mathbf{Z}_2\mathbf{H}_B)^\top.$$

Because $\mathbf{H}_B$ is symmetric ($\mathbf{H}_B = \mathbf{H}_B^\top$) and idempotent ($\mathbf{H}_B^2 = \mathbf{H}_B$), this expression simplifies to:

$$\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right) = \frac{1}{B}\mathbf{Z}_1\mathbf{H}_B\mathbf{Z}_2^\top,$$

completing the proof. □

Proof of Proposition 6.1. Recall the definition of Matrix KL divergence:

$$\mathrm{MKL}(\mathbf{P} \,||\, \mathbf{Q}) = \mathrm{tr}(\mathbf{P}\log\mathbf{P} - \mathbf{P}\log\mathbf{Q} - \mathbf{P} + \mathbf{Q}),$$

Substitute $\mathbf{P} = \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top$ and $\mathbf{Q} = \frac{1}{d}\mathbf{I}_d$ into this:

$$\begin{aligned}
\mathrm{MKL}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top \,\middle|\middle|\, \frac{1}{d}\mathbf{I}_d\right) &= \mathrm{tr}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top \log\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) - \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top \log\left(\frac{1}{d}\mathbf{I}_d\right) - \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \frac{1}{d}\mathbf{I}_d\right) \\
&= \mathrm{tr}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top \log\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) + \frac{\log d}{B}\mathbf{Z}\mathbf{Z}^\top - \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top + \frac{1}{d}\mathbf{I}_d\right) \\
&= -\,\mathrm{VNE}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) + \frac{\log d}{B}\,\mathrm{tr}(\mathbf{Z}\mathbf{Z}^\top) - \frac{1}{B}\,\mathrm{tr}(\mathbf{Z}\mathbf{Z}^\top) + \frac{1}{d}\,\mathrm{tr}(\mathbf{I}_d) \\
&= -\,\mathrm{VNE}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) + \log d - 1 + \frac{d}{d} \\
&= -\,\mathrm{VNE}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) + \log d,
\end{aligned}$$

From this, we conclude that:

$$\mathrm{VNE}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) = -\,\mathrm{KL}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top \,||\, \frac{1}{d}\mathbf{I}_d\right) + \log d.$$

$$\mathrm{ME}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) = \mathrm{VNE}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) + \mathrm{tr}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) = \mathrm{VNE}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) + 1.$$

The effective rank is defined as:

$$\mathrm{erank}(\mathbf{A}) = \exp\{\mathrm{H}(p_1, p_2, \ldots, p_n)\},$$

If we substitute $\mathbf{A} = \frac{1}{B}\mathbf{Z}\mathbf{Z}^\top$ and given that $\mathrm{VNE}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right)$ is the entropy of the eigenvalue distribution of $\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top$, then we could directly relate $\mathrm{erank}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right)$ and $\mathrm{VNE}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right)$:

$$\mathrm{erank}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) = \exp\left\{\mathrm{VNE}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right)\right\} = \exp\left\{\mathrm{ME}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) - 1\right\}.$$

Finally, we have

$$\mathrm{erank}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top\right) = \exp\left(\log d - \mathrm{MKL}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top \,\middle|\middle|\, \frac{1}{d}\mathbf{I}_d\right)\right) = \frac{d}{\exp\left(\mathrm{MKL}\left(\frac{1}{B}\mathbf{Z}\mathbf{Z}^\top \,||\, \frac{1}{d}\mathbf{I}_d\right)\right)}$$

Theorem A.1 (Taylor series expansion (Hall, 2013))**.** *The function*

$$\log\mathbf{A} = \sum_{m=1}^{\infty}(-1)^{m+1}\frac{(\mathbf{A}-\mathbf{I})^m}{m},$$

is defined and continuous on the set of all $n \times n$ complex matrices $\mathbf{A}$ with $\|\mathbf{A}-\mathbf{I}\| < 1$. For all $\mathbf{A}$ with $\|\mathbf{A}-\mathbf{I}\| < 1$,

$$e^{\log\mathbf{A}} = \mathbf{A}.$$

For all $\mathbf{X}$ with $\|\mathbf{X}\|_F < \log 2$, $\left\|e^{\mathbf{X}} - \mathbf{I}\right\| < 1$ and

$$\log e^{\mathbf{X}} = \mathbf{X}.$$

B. Details on Experiments

B.1. More on Loss Functions

Now we take a closer look at the loss function:

$$\begin{aligned}
\mathcal{L}_{\text{Matrix-SSL}} &= \mathcal{L}_{\text{Matrix-Uniformity}} + \mathcal{L}_{\text{Matrix-Alignment}(\gamma)} \\
&= \operatorname{MCE}\left(\frac{1}{d}\mathbf{I}_d, \mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right) - \operatorname{tr}\left(\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right) + \gamma \cdot \operatorname{MCE}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1), \mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right) \\
&= -\operatorname{tr}\left(\left(\frac{1}{d}\mathbf{I}_d\right)\log\left(\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right)\right) - \gamma \cdot \operatorname{tr}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1)\log\left(\mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right)\right) + \gamma \cdot \operatorname{tr}\left(\mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right) + \text{Const.}
\end{aligned} \tag{17}$$

Employing matrix KL divergence. As we previously introduced in Section 7, applying the stop gradient technique to the first branch $\mathbf{Z}_1$, as utilized in SimSiam (Hua, 2021), renders the third term $\operatorname{ME}(\operatorname{C}(\mathbf{Z}_1, \mathbf{Z}_1))$ a constant in the Matrix-Alignment-KL loss, as delineated in Equation 18.

$$\begin{aligned}
\mathcal{L}_{\text{Matrix-Alignment-KL}} &= -\operatorname{tr}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_2)\right) + \gamma \cdot \operatorname{MKL}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1)||\mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right) \\
&= -\operatorname{tr}\left(\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right) + \gamma \cdot \operatorname{MCE}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1), \mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right) - \gamma \cdot \operatorname{ME}(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1)).
\end{aligned} \tag{18}$$

$$\begin{aligned}
\mathcal{L}_{\text{Matrix-SSL-KL}} &= \mathcal{L}_{\text{Matrix-Uniformity-KL}} + \mathcal{L}_{\text{Matrix-Alignment-KL}(\gamma)} \\
&= \operatorname{MKL}\left(\frac{1}{d}\mathbf{I}_d||\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right) - \operatorname{tr}\left(\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right) + \gamma \cdot \operatorname{MKL}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1)||\mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right) \\
&= \operatorname{MKL}\left(\frac{1}{d}\mathbf{I}_d||\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right) - \operatorname{tr}\left(\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right) + \gamma \cdot \operatorname{MCE}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1), \mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right) - \gamma \cdot \operatorname{ME}(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1)) \\
&= \operatorname{MCE}\left(\frac{1}{d}\mathbf{I}_d, \mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right) - \operatorname{ME}\left(\frac{1}{d}\mathbf{I}_d\right) - \operatorname{tr}\left(\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right) \\
&\quad + \gamma \cdot \operatorname{MCE}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1), \mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right) - \gamma \cdot \operatorname{ME}(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1)) \\
&= -\operatorname{tr}\left(\left(\frac{1}{d}\mathbf{I}_d\right)\log\left(\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right)\right) + \operatorname{tr}\left(\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right) + \text{Const.} \ - \operatorname{tr}\left(\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right) \\
&\quad - \gamma \cdot \operatorname{tr}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1)\log\left(\mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right)\right) + \gamma \cdot \operatorname{tr}\left(\mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right) - \gamma \cdot \operatorname{ME}(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1)) \\
&= -\operatorname{tr}\left(\left(\frac{1}{d}\mathbf{I}_d\right)\log\left(\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right)\right) - \gamma \cdot \operatorname{tr}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1)\log\left(\mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right)\right) \\
&\quad + \gamma \cdot \operatorname{tr}\left(\mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right) + \gamma \cdot \operatorname{ME}(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1)) + \text{Const.} \qquad \left(\xRightarrow{\text{Stop Gradient on } \mathbf{Z}_1}\right) \\
&= -\operatorname{tr}\left(\left(\frac{1}{d}\mathbf{I}_d\right)\log\left(\mathbf{C}\left(\mathbf{Z}_1, \mathbf{Z}_2\right)\right)\right) - \gamma \cdot \operatorname{tr}\left(\mathbf{C}(\mathbf{Z}_1, \mathbf{Z}_1)\log\left(\mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right)\right) + \gamma \cdot \operatorname{tr}\left(\mathbf{C}(\mathbf{Z}_2, \mathbf{Z}_2)\right) + \text{Const.}
\end{aligned} \tag{19}$$

From Equation 17 and 19, we find that they are essentially the same loss function when the stop gradient is performed.

C. Neural Collapse and Dimensional Collapse

Feature representations acquired through a deep neural network employing a cross-entropy (CE) loss optimized by stochastic gradient descent, are capable of attaining zero loss (Du et al., 2018) with arbitrary label assignments (Zhang et al., 2021). A phenomenon which known as neural collapse (NC) (Papyan et al., 2020) is observed when training of the neural network continues beyond zero loss with CE. Galanti et al. (2021) demonstrate that the NC phenomenon can facilitate some transfer learning tasks. However, potential concerns associated with neural collapse exist, as Ma et al. (2023) posit that the total within-class features collapse may not be ideal for fine-grained classification tasks.

The NC phenomenon embodies the following characteristics (Zhu et al., 2021):

- Variability collapse: The intra-class variability of the final layer's features collapse to zero, signifying that all the features of a single class concentrate on the mean of these features for each class respectively.

- Convergence to Simplex ETF: Once centered at their global mean, the class-means are simultaneously linearly separable and maximally distant on a hypersphere. This results in the class-means forming a simplex equiangular tight frame (ETF), a symmetrical structure determined by a set of points on a hypersphere that is maximally distant and equiangular to each other.
- Convergence to self-duality: The linear classifiers, existing in the dual vector space of the class-means, converge to their respective class-mean and also construct a simplex ETF.
- Simplification to Nearest Class-Center (NCC): The linear classifiers behaviors similarly to the nearest class-mean decision rule.

Here we present the definition of standard K-Simplex ETF and general K-Simplex ETF (Papyan et al., 2020).

Definition C.1 (K-Simplex ETF)**.** A standard Simplex ETF is characterized as a set of points in $\mathbb{R}^K$, defined by the columns of

$$\mathbf{M} = \sqrt{\frac{K}{K-1}}\left(\mathbf{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right),$$

where $\mathbf{I}_K \in \mathbb{R}^{K\times K}$ is the identity matrix, and $\mathbf{1}_K \in \mathbb{R}^K$ represents a all-one vector. Consequently, we also obtain

$$\mathbf{M}^\top\mathbf{M} = \mathbf{M}\mathbf{M}^\top = \frac{K}{K-1}\left(\mathbf{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right).$$

Definition C.2 (General K-Simplex ETF)**.** A general Simplex ETF is characterized as a set of points in $\mathbb{R}^K$, defined by the columns of

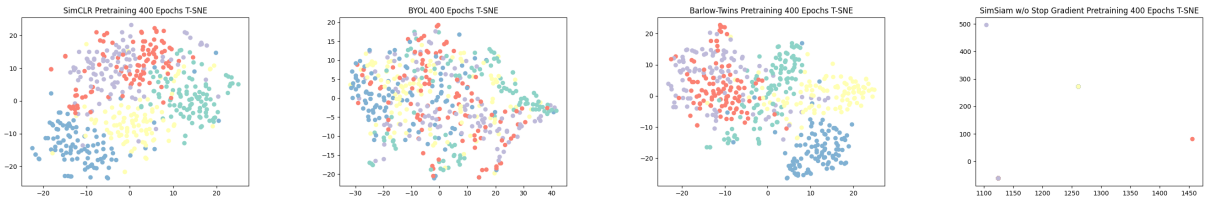
$$\tilde{\mathbf{M}} = \alpha\mathbf{U}\mathbf{M},$$

where $\alpha \in \mathbb{R}_+$ is a scale factor, and $\mathbf{U} \in \mathbb{R}^{p\times K}$ $(p \geq K)$ is a partial orthogonal matrix $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$.

Zhu et al. (2021) further studied the problem using an unconstrained feature model that separates the topmost layers from the classifier of the neural network. They established that the conventional cross-entropy loss with weight decay presents a benign global landscape, where the only global minimizers are the Simplex ETFs and all other critical points are strict saddles exhibiting negative curvature directions.

The study was later extended (Zhou et al., 2022), demonstrating through a global solution and landscape analysis that a wide range of loss functions, including commonly used label smoothing (LS) and focal loss (FL), display Neural Collapse. Therefore, all pertinent losses (i.e., CE, LS, FL, MSE) yield comparable features on training data.

D. Measuring Dimensional Collapse



(a) SimCLR (b) BYOL (c) Barlow Twins (d) SimSiam (collapsed w/o stop gradient)

Figure 2. Visualization of feature representation for images in 5 different classes from CIFAR-100 dataset via t-SNE of various self-supervised learning methods. We find that SimCLR has larger inter-class variability than others, as the clusters seem more separable. For illustration, we also introduce a collapsed representation via SimSiam without stop gradient operation.

Papyan et al. (2020) discuss the fascinating occurrence of neural collapse during the training of a supervised neural network utilizing cross-entropy loss for classification tasks that result in an intra-class collapse. Contrastive learning has effects of dimensional collapse due to its spectral clustering nature (Tan et al., 2023b). As dimension-contrastive learning can be seen

as pursuing uniformity, we are also interested in discovering the relationship between dimension-contrastive learning and dimensional collapse.

Figure 2 illustrates that the non-contrastive method, Barlow Twins, exhibits greater intra-class variability than the contrastive method, SimCLR. However, for larger samples and classes (e.g., Figure 4 in Appendix C), this observation is qualitative explicit. To quantify this observation, we propose the introduction of metrics involving class-specific information to quantify dimensional collapse. These measures may enhance our understanding of the differences among supervised learning, contrastive, and non-contrastive SSL.

Assuming a total of K classes and n labeled samples $\{x_i, y_i\}_{i=1}^n$, denote the number of samples in each class c as n_c, i.e., $n_c = |\{i \mid y_i = c\}|$. We define the *intra-class effective rank* and *inter-class effective rank* as follows.

Definition D.1 (Intra-class effective rank)**.** Denote the class-mean vector of each class c as $\mu_c = \frac{1}{n_c} \sum_{y_i=c} \mathbf{f}(\mathbf{x}_i)$, and denote $\mathbf{C}(\mathbf{f}(x) \mid y)) = \frac{1}{n_y} \sum_{y_i=y} (\mathbf{f}(x_i) - \mu_y)(\mathbf{f}(x_i) - \mu_y)^\top$. We define *intra-class effective rank* (intra-class erank) as

$$\text{erank}_{\text{intra-class}} \triangleq \frac{1}{K} \sum_{y \in [K]} \text{erank}(\mathbf{C}(\mathbf{f}(\mathbf{x}) \mid y))), \tag{20}$$

which can be viewed as an empirical approximation of $\mathbb{E}_{y \in [K]} [\text{erank}(\mathbf{C}(\mathbf{f}(x) \mid y))]$, where x is drawn from p_{data}.

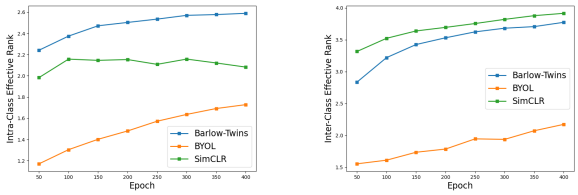
Definition D.2 (Inter-class effective rank)**.** Denote global mean of representation as $\mu_G = \frac{1}{n} \sum_{i \in [n]} \mathbf{f}(x_i)$, then we define *inter-class effective rank* (inter-class erank) as the effective rank of the covariance matrix of all C class-mean vectors,

$$\text{erank}_{\text{inter-class}} \triangleq \text{erank}[\frac{1}{K} \sum_{i \in [K]} (\mu_i - \mu_G)(\mu_i - \mu_G)^\top]. \tag{21}$$

When class are balanced, intra-class erank is approximately $\text{erank}(\mathbf{C}_{y \in [K]}(E[\mathbf{f}(x) \mid y]))$, where x is drawn from p_{data}.

Remark. These two metrics can be interpreted as an effective rank factorization of the two terms in the total covariance theorem.

From illustrative examples shown in Figure 3, we observe that SimCLR, as a contrastive method, exhibits a consistent decrease in intra-class effective rank during training. This empirical evidence corroborates the spectral clustering interpretation of contrastive learning. On the contrary, non-contrastive methods like BYOL and Barlow Twins, owing to the inherent property of kernel-uniformity loss (and its low-order Taylor approximations) tending towards a uniform distribution, exhibit larger intra-class effective ranks that continue to increase during training. Regarding the inter-class effective rank, a metric for global class-means effective rank, all three methods show a consistent increase.



(a) Intra-class erank on test dataset (b) Inter-class erank on test dataset

Figure 3. Intra-class effective rank and inter-class effective rank. It is obvious that intra-class effective rank continues to grow for BYOL or Barlow Twins, but not for SimCLR.

We now present some theoretical properties of effective rank and its connections to an equiangular tight frame (ETF). The following theorem suggests that a larger effective rank of the Gram matrix is beneficial for expressiveness.

Theorem D.3 (Maximize effective rank forms a equiangular tight frame (ETF))**.** *For K vectors $\mathbf{z}_i$ ($1 \leq i \leq K$), each lying on S^{d-1}. Assuming the latent dimension d satisfies $d \geq K$ and the mean of $\mathbf{z}_i$ is $\mathbf{0}$, denote $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_K]$. If the Gram matrix $\mathbf{Z}^\top \mathbf{Z}$ has an effective rank of $K-1$, it implies the existence of an equiangular tight frame (ETF) in the orthonormal span of $\mathbf{z}_i$. Conversely, the Gram matrix of any ETF has an effective rank of $K-1$.*

Proof. Since the mean vector is $\mathbf{0}$, the Gram matrix can have an effective rank of at most $K-1$. By Property 1 in (Roy & Vetterli, 2007), we deduce that the Gram matrix $\mathbf{Z}^\top \mathbf{Z}$ has $K-1$ equal eigenvalues and one eigenvalue equal to 0.

The trace of the Gram matrix equals K because $\mathbf{z}_i$ lies on S^{d-1}. Hence, the Gram matrix has $K-1$ equal eigenvalues of $\frac{K}{K-1}$ and one eigenvalue of 0. Therefore, the Gram matrix shares the same eigenvalues (spectrum) as $\frac{K}{K-1}\mathbf{H}_K$, where $\mathbf{H}_K$ is the centering matrix $\mathbf{I}_K - \frac{1}{K}\mathbf{1_K}\mathbf{1_K}^\top$.

Given the orthonormal standard form, there exists an orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{K\times K}$ such that $\mathbf{Q}^\top(\mathbf{Z}^\top\mathbf{Z})\mathbf{Q} = \frac{K}{K-1}\mathbf{H}_K$. According to Lemma 11 in Papyan et al. (2020), $\mathbf{ZQ}$ constitutes an ETF. As $\mathbf{ZQ}$ directly represents the orthonormal span of $\mathbf{Z}$'s column space, the conclusion follows. □

Gram matrix plays a key role in connecting our metric with Section 6, i.e., understanding the rank-increasing phenomenon.

Theorem D.4. *The effective rank of the total sample Gram matrix can be effectively estimated by batch.*

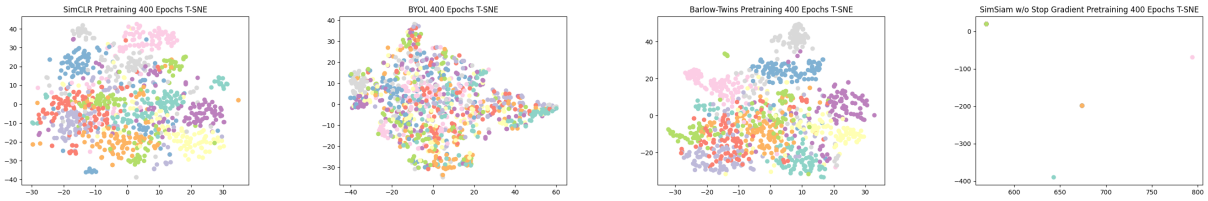
Proof. Note scaling does not change effective rank. Change the order of $\mathbf{Z}^\top\mathbf{Z}$ to $\mathbf{ZZ}^\top$, then can rewrite self-correlation as the empirical estimation of expected self-correlation by samples in a batch. This explains the estimation given by Zhuo et al. (2023). □

Interestingly, the following theorem connects our metrics with the Gram matrix.

Theorem D.5. *Assuming the dataset is class-balanced and the global mean is* 0*, then the effective rank of the covariance matrix of all K class-mean vectors is exactly the same as the effective rank of the Gram matrix.*

Proof. As $\mathbf{ZZ}^\top$ and $\mathbf{Z}^\top\mathbf{Z}$ have the same non-zero eigenvalues, thus having the same effective rank. □

D.1. Experiments on Dimensional Collapse



(a) SimCLR (b) BYOL (c) Barlow Twins (d) SimSiam (collapsed w/o stop gradient)

Figure 4. Visualization of feature representation for images in 10 different classes from CIFAR-100 dataset via t-SNE of various self-supervised learning methods. We find that in many categories, it is difficult to distinguish between two non-contrastive methods (BYOL, Barlow Twins) and contrastive method (SimCLR) by t-SNE.

We measure dimensional collapse on various self-supervised learning methods, including SimCLR (Chen et al., 2020a), BYOL (Grill et al., 2020), Barlow Twins (Zbontar et al., 2021) and SimSiam (Chen & He, 2021) with or without stop gradient. We reproduce the above methods on the self-supervised learning task of CIFAR100 (Krizhevsky et al., 2009) dataset, using the open source implementations (Tsai et al., 2021a; Hua, 2021) of the above methods tuned for CIFAR. After pre-training, we use the saved checkpoints to evaluate the results of these methods on different metrics.

We calculate the intra-class and inter-class effective rank directly by definition, while for MCE, we shuffle the testing dataset, import the data with 512 batch size, and finally output the average metrics of all batches.

We perform t-SNE (van der Maaten & Hinton, 2008) visualization on the last checkpoint of each method with the help of scikit-learn (Pedregosa et al., 2011). We use the default t-SNE (van der Maaten & Hinton, 2008) parameter of scikit-learn (Pedregosa et al., 2011) and select the first 5 or 10 categories from 100 categories in CIFAR-100 (Krizhevsky et al., 2009) for visualization.