Understanding Contrastive Learning via Gaussian Mixture Models

Parikshit Bansal
UT Austin
pbansal@utexas.edu

Ali Kavis
UT Austin
kavis@austin.utexas.edu

Sujay Sanghavi
UT Austin
sanghavi@mail.utexas.edu

Abstract

Contrastive learning involves learning representations via a loss function that encourages each (unlabeled) sample to be far from other samples, but close to its own *augmentation*. In this paper, we aim to understand why this simple idea performs remarkably well, by theoretically analyzing it for a simple, natural problem setting: dimensionality reduction in Gaussian Mixture Models (GMMs). Note that the standard GMM setup lacks the concept of augmentations. We study an intuitive extension: we define the pair of data sample and its augmentation as a coupled random draw from the GMM such that the marginal over the "noisy" augmentation is *biased* towards the component of the data sample. For this setup, we show that vanilla contrastive loss, e.g., InfoNCE, is able to find the *optimal* lower-dimensional subspace even when the Gaussian components are non-isotropic. In particular, we show that InfoNCE can match the performance of a fully supervised algorithm, e.g., LDA, (where each data point is labeled with the mixture component it comes from) even when the augmentations are "noisy". We further extend our setup to the multi-modal case, and develop a GMM-like setting to study the contrastive CLIP loss. We corroborate our theory with experiments on CIFAR100; representations learned by InfoNCE loss match the performance of LDA on clustering metrics.

1 Introduction

Contrastive learning (CL) is now a gold-standard paradigm for learning representations with popular examples such as vision models like SimCLR [6] and MoCo [8] , text models like Dense Passage Retrieval (DPR) [18], and vision-language models like CLIP [20]. Once learned, these representations both demonstrate commendable zero-shot performance, and could easily be adapted to achieve SoTA performance on various tasks like classification [6] and retrieval [13].

Contrastive learning is a self-supervised strategy that aims to learn coherent representations given an unlabeled dataset. The core idea is to relate *similar* (positive) data samples to each other while contrasting *unrelated* (negative) points. To formalize, consider a pair of points $(\boldsymbol{x}, \hat{\boldsymbol{x}})$, where $\boldsymbol{x}$ is a sample from the dataset and $\hat{\boldsymbol{x}}$ is a "similar", positive example. Contrastive methods learn representations so that the embedding of the sample $\boldsymbol{x}$ is *close* to that of its partner $\hat{\boldsymbol{x}}$, while being *far away* from other points. Sometimes, these pairs are naturally present. For instance, in CLIP, each training data comes as a pair of an image and the corresponding text caption. In other scenarios, they can be manually obtained via data augmentation that preserves the underlying semantics, e.g., we can define an image and its resized/cropped/rotated version as a pair.

It is fundamentally important to understand **why this simple idea of relating/contrasting *pairs* works so well**. In this paper, we aim to answer this question with a theoretical study of contrastive learning by casting is as dimensionality reduction for Gaussian Mixture Models (GMMs). We motivate our study through the lens of representation learning which aims at finding a mapping between the high-dimensional input space, e.g., images of 1 million pixels, and a lower dimensional

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

output domain of *representations*, e.g., 1024-dimensional feature vectors. Similarly, contrastive learning aims at discovering robust mappings between the data and the representations by exploiting additional information in the form of augmentations. In our framework of GMMs, we consider *linear* mappings/projections (for analytical tractability) and characterize the *optimality* of the mapping functions learned by contrastive methods by analyzing the associated projection subspaces.

To highlight the success of contrastive methods in learning linear projectors, we conduct our study in comparison to fully supervised methods and standard spectral methods, i.e., based on singular value decomposition (SVD). We provide an example where we provide a three-way comparison in finding anticipated projection subspaces (see Section 3.1). Note that SVD-based methods are not designed to leverage augmentations. Therefore, we use SVD-based methods merely as a reference point to argue that *when (noisy) augmentations are present*, contrastive learning can *provably* go beyond what would otherwise be possible. However, a more relevant and interesting comparison is with respect to supervised methods, i.e., LDA in this case. In that sense, we provide principled, theoretical insight into how contrastive learning works and how it performs with respect to a measure of optimality.

Within our GMM framework, a key proposal of ours is formalizing the "notion of augmentations", i.e. point-pairs, going beyond the idealized definition considered in [3]. In the context of GMMs, a natural choice of augmentation would be drawing a random sample *from the same component* as the original point belongs to. Instead, we assume that for every point $\boldsymbol{x}$, its augmentation $\hat{\boldsymbol{x}}$ is a random draw *from the GMM* following a distribution (see Definition 2.2) that is *biased towards the component of the original point* $\boldsymbol{x}$. To keep the setting realistic, we are oblivious to which GMM component either of them comes from.

Given the model of noisy augmentations, we analyze contrastive learning objectives in two categories; single and multi-modal GMMs. In the single modal setting, the points in a pair are drawn from the same GMM distribution. We analyze InfoNCE and SimSiam losses, which are suited to leverage the additional information provided by the augmentations. In the **multi-modal** context, the pair $(\boldsymbol{x}, \hat{\boldsymbol{x}})$ will be generated by *separate* GMMs. For instance, in the CLIP [] images are associated with text descriptions such that we have pairs of the form $(\boldsymbol{x}_V, \boldsymbol{x}_T)$, where $\boldsymbol{x}_V$ (images) and $\boldsymbol{x}_T$ (texts) can have different dimensions. The objective now is to learn two different projections, one for each modality to project them onto the *same* space. Contrastive methods aim to make the embedding of $\boldsymbol{x}_V$ close to that of its pair $\boldsymbol{x}_T$, and far from random other $\tilde{\boldsymbol{x}}_T$. Since the samples come in natural pairs, we do not need augmentations. We provide further details in Section 5.

Contributions: Our objective is to theoretically understand the vital role of augmentations in self-supervised representation learning; we do so in the context of GMMs. To summarize,

1. We formalize a generalized notion of **noisy augmentation/point pairs**, enabling us to concretely characterize contrastive learning methods in the context of GMMs.
2. We quantify a measure of optimality for the projections learned in the context of GMMs. We show that InfoNCE can find optimal projections for GMMs with *general shared covariances*.
3. For the multi-modal setting, we show that the CLIP loss learns linear projections that are a subset of the (Fisher-)optimal subspaces for each modality, filtering out noise directions.

1.1 Related work

Contrastive learning and InfoNCE. Contrastive learning is at the heart of state-of-the-art representation learning methods like SimCLR [6], MoCo [8] etc. Prior theoretical work on contrastive learning argue that the contrastive objective leads to learning of "general purpose" representations that facilitates better downstream performance and sample complexity. [2] gives the first theoretical analyses of the contrastive loss, showing provable guarantees on downstream task with reduced sample complexity. [12] relaxes the augmentation assumptions made in [2] by showing that contrastive learning is equivalent to spectral clustering on some appropriately defined data graph. [22, 11] extend this line of work by arguing for the inclusion of the inductive biases of the neural network architecture into the theoretical analyses. [16, 27] extend the setup of [12] to show that the eigenfunctions of the positive pair graph (as defined in [12]) are the basis of the set of view-invariant functions. They further show that these representations are equivalent to Kernel PCA with "positive-pair kernel".

Self-supervised learning. BYOL [10], SwAV [5], SimSiam [7] are some of the most popular methods for self-supervised representation learning. They are also based on Siamese architecture

used in contrastive methods, but notably, they do not make use of *negatives* to prevent "collapsing" solutions. Contrastive methods like SimCLR [6] and CLIP [20] aim to learn representations so that embedding of a point is close to its partner while far away from that of any other point. On the contrary, *non-contrastive* self-supervised learning methods only enforce the former condition and do not explicitly deal with the separation between the embeddings of samples. These methods are empirically motivated by the notion of breaking the symmetry between the encoders in Siamese networks [25]. [23, 15, 26] propose various techniques for breaking this symmetry and study their training dynamics to understand their role in preventing collapse. Compared to the relevant work on self-supervised learning, we instead focus on the fixed point analysis of the loss functions.

Dimensionality reduction for GMMs. Spectral methods are popularly used for dimensionality reduction in GMMs by analyzing the principal directions of the data covariance. Spectral clustering algorithms [21, 24, 1, 17, 4] essentially combine spectral analysis with standard clustering algorithms (e.g. K-Means) in the low dimensional space. Since their error bounds grow as square root of the ambient dimension, the two-step approach leads to lower clustering error than in the original space.

2 Problem setup

We consider the task of learning a "good" representation function for points $\boldsymbol{x}$ drawn from a data distribution. Informally, such a function should map the data onto a lower-dimensional subspace while preserving the class information. For tractability, we restrict f to the class of linear mappings; $\boldsymbol{x} \mapsto \boldsymbol{A}^T\boldsymbol{x}$, where $\boldsymbol{A} \in \mathbb{R}^{d\times r}$. In this setup, we will show that self-supervised methods can learn the optimal mapping for this task. We compare and contrast our results for self-supervised methods to well-known results for supervised methods (e.g., LDA) and spectral methods.

As the basis of our study, we consider data distributions following a Gaussian mixture model (GMM). Intuitively, each mixture component in GMM represents a class in data and we assume that components have a shared covariance.

Definition 2.1. A *Shared Covariance Gaussian Mixture Model* (SharedGMM) parameterized by $\{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}\}_{k\in[K]}$ is defined as the probability distribution $F = \sum_{k\in[K]} w_k\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$ where $\sum_k w_k = 1$, $\boldsymbol{\mu}_k$ are the means and $\boldsymbol{\Sigma} \in \mathbb{S}^d_{++}$ is the covariance matrix shared by the components.

To complete our data model, we formalize the definition of *augmentation*. Given a point $\boldsymbol{x}$, its augmentation $\hat{\boldsymbol{x}}$ is an independent sample from the mixture distribution with a *bias* towards the underlying component of $\boldsymbol{x}$. The bias implies that if $\boldsymbol{x}$ is sampled from a component z, then its augmentation, on average, is more likely to be sampled from the same component z.

Definition 2.2. For a SharedGMM F parameterized by $\{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}\}_{k\in[K]}$, we define its *Augmentation-enabled Distribution* (AeD) $\hat{F}$, parameterized by $\{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}, \delta\}_{k\in[K]}$ as

$$\hat{F} = \delta \sum\nolimits_k w_k\big(\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \times \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})\big) + (1-\delta)\big(\sum\nolimits_k w_k\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \times \sum\nolimits_{k'} w_{k'}\mathcal{N}(\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma})\big)$$

To elucidate, augmentation-enabled distribution (AeD) returns a pair of points $(\boldsymbol{x}, \hat{\boldsymbol{x}}) \sim \hat{F}$ in a two-step process. We first flip a coin with bias δ. Based on the coin flip, we either sample twice from the same component or sample from possibly different components (chosen with probability w_k). The bias of the coin δ controls the correlation between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$: $\delta = 0$ means $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are independent samples, while $\delta = 1$ means $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are independent draws from the same component. The definition ensures that the marginal of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are equal to F for any δ.

Spectral versus self-supervised methods. Our results corroborate the known advantages of self-supervised methods over spectral methods. Clearly, spectral methods learn linear mappings directly on the data samples $\boldsymbol{x} \sim \mathcal{D}$, whereas, self-supervised methods operate on the augmented pairs $(\boldsymbol{x}, \hat{\boldsymbol{x}}) \sim \mathcal{D}_{\text{pair}}$. Consequently, their optimization problems differ in their target objectives:

$$\mathcal{L}_{\text{spectral}}(\boldsymbol{A}; \mathcal{D}) = -||\boldsymbol{A}^T\boldsymbol{x}||^2 \qquad \& \qquad \mathcal{L}_{\text{selfsup}}(\boldsymbol{A}; \mathcal{D}_{\text{pair}}) = -\underbrace{(\boldsymbol{A}^T\boldsymbol{x})^T\boldsymbol{A}^T\hat{\boldsymbol{x}}}_{\text{attractive term}} + \underbrace{\mathcal{R}(\boldsymbol{x})}_{\text{regularizer}}$$

While spectral objectives find the *best-fit* subspace explaining maximum data variance, self-supervised objectives are based on the principle of bringing embeddings of similar point closer in conjunction with a loss-specific regularizer. The regularizer typically induces separation between distinct points.

Linear dimensionality reduction (LDR). Framing our task as LDR allows us to establish a measure of comparison between mappings learned by spectral ($\boldsymbol{A}_{\text{spectral}}$) and self-supervised methods ($\boldsymbol{A}_{\text{selfsup}}$). For a SharedGMM, we denote it's "intra-component" variance (i.e. variance within a component) by $\boldsymbol{\Sigma}$ and "inter-component" variance (i.e. the separation between components) by $\sum_k w_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T$ [9]. A GMM is said to be *well-separated* if it has *low intra-component* variance and *high inter-component* variance. LDR aims to map a GMM to a low dimensional subspace where it is well-separated. We evaluate the performance of these mappings by Fisher discriminant [9].

Definition 2.3 (Fisher Discriminant)**.** Suppose a SharedGMM (Def 2.1) parameterized by $\{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}\}_{k\in[K]}$. Then, the *fisher discriminant* $J(\boldsymbol{A})$ for a mapping $\boldsymbol{A}$ is defined as:

$$J(\boldsymbol{A}) = \operatorname{Tr}\left(\left[\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}\right]^{-1}\left[\boldsymbol{A}^T\left(\sum\nolimits_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T\right)\boldsymbol{A}\right]\right)$$

Remark 2.4. *While $J(\boldsymbol{A})$ is defined in terms of $\boldsymbol{A}$, its value is solely a function of $\operatorname{Col}(\boldsymbol{A})$, i.e. the column space of $\boldsymbol{A}$. We hence use the mapping matrix $\boldsymbol{A}$ and projection subspace $\operatorname{Col}(\boldsymbol{A})$ interchangeably in our discussion.*

Note that $\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A}$ and $\boldsymbol{A}^T(\sum_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T)\boldsymbol{A}$ denote the intra-component and the inter-component variances, respectively, *after* projection via $\boldsymbol{A}$. Thus, the Fisher discriminant measures the ratio of inter-component to intra-component variances post-projection. A "favorable" mapping $\boldsymbol{A}$ will have a high value of the Fisher discriminant $J(\boldsymbol{A})$, hence, the Fisher discriminant J serves as the quantitative metric for evaluating the subspaces learned by different frameworks.

Connection to linear discriminant analysis (LDA). One can argue that it is not surprising for self-supervised methods to have better performance than classical spectral methods. The more interesting comparison would be against a *supervised* method, which would help uncover the capabilities of contrastive methods. Therefore, we focus our attention on Linear Discriminant Analysis (LDA) [9]. LDA is a *supervised* LDR method that leverages the class information to learn the subspace where the data distribution conditioned on the class label (i.e. underlying component index) is maximally separated (see Appendix H). We highlight that assuming AeD (Def. 2.2) is strictly weaker than assuming a *labeled* dataset. We show that this relaxed condition is enough for learning the Fisher subspace and gives similar performance to supervised LDR methods.

3 Fisher subspace

For any target dimension r, the linear map that has the top r eigenvectors of $\boldsymbol{\Sigma}^{-1}(\sum_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T)$ as its columns maximizes the Fisher discriminant. Similarly, for any mappings $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, if $\operatorname{Col}(\boldsymbol{A}_1) \subseteq \operatorname{Col}(\boldsymbol{A}_2)$, then we have $J(\boldsymbol{A}_1) \leq J(\boldsymbol{A}_2)$. In other words, $J(\cdot)$ is monotonic in the subspace of the matrices. Based on this, we will have the following definition, which is necessary in *measuring the optimality of projections* learned by different methods.

Definition 3.1 (Fisher Subspace)**.** Given a SharedGMM (Def 2.1) parameterized by $\{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}\}_{k\in[K]}$, its *Fisher subspace* [9], denoted by S_F, is the smallest subspace that achieves the maximum Fisher discriminant, which is given by:

$$S_F = \operatorname{Span}\big(\{\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k\}_{k\in[K]}\big) \tag{1}$$

We also use a unique property of the Fisher subspace [9]; it is the smallest subspace preserving the class posterior probabilities for Gaussian mixtures with shared covariance, which we formalize next.

Lemma 3.2. *Let $\{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}\}_{k\in[K]}$ be a SharedGMM and $\Pr(z = k|\boldsymbol{x})$ be the posterior probability of $\boldsymbol{x}$ being drawn from the component z. Let S_F be the mixture's Fisher subspace and $\boldsymbol{A}_F$ be a projection matrix such that $\operatorname{Col}(\boldsymbol{A}_F) = S_F$. Then for any $\boldsymbol{x} \sim F$,*

$$\Pr(z = k|\boldsymbol{x}) = \Pr(z = k|\boldsymbol{A}_F^T\boldsymbol{x})\ \forall k \in [K]$$

Crucially, this property implies that projecting a GMM onto its Fisher subspace will not result in mode collapse or erroneous mode merging. Moreover, any clustering algorithm operating in the lower dimensional subspace will observe the same class probabilities as in the original space. Owing to these properties of the Fisher subspace, we argue that it is *the optimal subspace for projection*, particularly for the class of shared covariance GMMs. Under the fully labeled SharedGMM setting, one could deduce that the subspace learned by (multi-class) LDA, S_{LDA}, for SharedGMM coincides with Fisher subspace S_F.

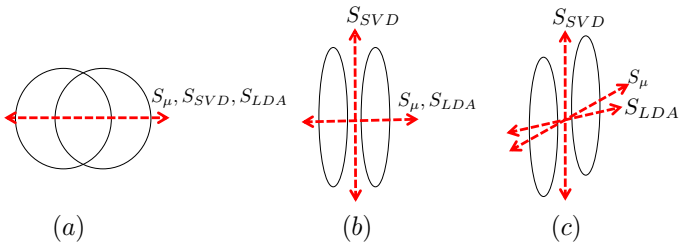


Figure 1: (a) (Spherical Gaussians) $S_{SVD} = S_{LDA}$ and projection onto S_{SVD} leads to *well seperated* GMM. (b) (Non-spherical Gaussians) Large variance in one direction means $S_{SVD} \neq S_{LDA}$ and projection onto S_{SVD} leads to mode collapse. (c) (Shifted non-spherical Gaussians) $S_{\boldsymbol{\mu}} \neq S_{LDA}$.

3.1 A simple example: Parallel pancakes [17]

Spectral methods [17] are the standard methods for LDR. They are based on the principle of finding the best-fit subspace which is (generally) given by the *top singular vectors of data covariance*. Formally, the r-dimensional *SVD subspace* [24] $S^{(r)}_{SVD}$ for a GMM F is described as the top r left singular vectors of $\mathbb{E}_{\boldsymbol{x}\sim F}[\boldsymbol{x}\boldsymbol{x}^T]$. A vanilla spectral dimensionality reduction method hence projects the data points onto the r-dimensional SVD subspace. [17] proved that for *spherical* GMMs (i.e. $\boldsymbol{\Sigma} = \boldsymbol{I}$), SVD subspace is equal to the mean subspace $S_{\boldsymbol{\mu}} = \mathrm{Span}(\boldsymbol{\mu})$. Moreover, using Eq. (1) we can derive that $S_{LDA} = S_{\boldsymbol{\mu}} = S_{SVD}$. Hence, projection onto the SVD subspace is *the optimal* dimensionality reduction method for *spherical* GMMs.

However, it is easy to construct examples demanding for a sophisticated method. Spectral methods aim to maximally explain the data, leading to possibly "bad" projections when the variance in certain directions dominates the separation between the means. In Figure 1 we present such an example; consider a two component GMM that resembles "parallel pancakes" such that the components are narrow and separated along one direction, and spherical in all other directions. The two dimensional SVD subspace for this GMM is given by a plane parallel to the pancakes. The Fisher subspace (and equivalently LDA subspace), however, is the plane *passing through the means*; formally, $S_{\mathrm{SVD}} \not\subset S_F$. In fact, the S_{SVD} has the *smallest* Fisher discriminant among all two-dimensional subspaces. Finding the optimal subspace for non-spherical GMMs is known to be non-trivial [4, 17].

4 Contrastive learning for Gaussian mixture models

We analyze solving the dimensionality reduction task for GMMs with two popular self-supervised methods, SimCLR and SimSiam. Both methods build on using augmentation pairs, but differ in their optimization objective (specifically how they regularize). We show that both objectives are able to effectively leverage the augmentations and learn mappings onto the (optimal) Fisher subspace.

4.1 Optimization objective

InfoNCE loss [19] is popularly used in contrastive learning methods (like SimCLR). It learns representations that pull data points and their augmentations close together while pushing them away from other points in the embedding space. We define the InfoNCE objective for linear mappings as,

$$\mathcal{L}_{\mathrm{Info}}(\boldsymbol{A}) = -\mathop{\mathbb{E}}_{(\boldsymbol{x},\hat{\boldsymbol{x}})\sim\hat{F}}\left[(\boldsymbol{A}^T\boldsymbol{x})^T\boldsymbol{A}^T\hat{\boldsymbol{x}}\right] + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim F}\left[\log\left(\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim F}\left[\exp\left((\boldsymbol{A}^T\boldsymbol{x})^T\boldsymbol{A}^T\tilde{\boldsymbol{x}}\right)\right]\right)\right]. \quad (2)$$

where $(\boldsymbol{x},\hat{\boldsymbol{x}}) \sim \hat{F}$ is a sample from augmentation-enabled distirbution (Def 2.2) such that $\boldsymbol{x} \sim F$ and $\tilde{\boldsymbol{x}} \sim F$ are independent draws from the mixture. The attractive term (i.e. the first term) keeps $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ close by maximizing their dot product, while regularization term penalizes proximity to random samples. In other terms, the first term increases inter-component variance while the second term decreases data variance and hence intra-component variance.

SimSiam loss [7] is another popular self-supervised loss without the negatives.

$$-\mathop{\mathbb{E}}_{(\boldsymbol{x},\hat{\boldsymbol{x}})\sim\hat{F}}\left[\left(\frac{\boldsymbol{A}_p^T\boldsymbol{A}^T\boldsymbol{x}}{||\boldsymbol{A}_p^T\boldsymbol{A}^T\boldsymbol{x}||_2}\right)^T \mathrm{StopGrad}\left(\frac{\boldsymbol{A}^T\hat{\boldsymbol{x}}}{||\boldsymbol{A}^T\hat{\boldsymbol{x}}||_2}\right)\right]$$

Unlike InfoNCE, Simsiam only operates on augmentations (from $\hat{F}$) and doesn't use negatives. The loss is parameterized by two matrices; the mapping $\boldsymbol{A}$ and the "prediction head" $\boldsymbol{A}_p$, which is not utilized for projection. Observe that Simsiam might be prone to a collapsing solution, i.e., mapping all points onto the same vector. Prior work [23, 15, 26] has argued that $\boldsymbol{A}_p$, which makes the optimization asymmetric, and StopGrad, which zeros out the gradients flowing through it, play a crucial role together in the guiding the training dynamics, preventing the occurrence of collapse. Our goal is to analyze the fixed point of this loss function. For tractability, we introduce some simplifying assumptions and examine a modified loss:

$$\mathcal{L}_{\text{Siam}}(\boldsymbol{A}) = -\,\mathbb{E}_{(\boldsymbol{x},\hat{\boldsymbol{x}})\sim\hat{F}}\left[(\boldsymbol{A}^T\boldsymbol{x})^T\boldsymbol{A}^T\hat{\boldsymbol{x}}\right] + \xi\,\mathbb{E}_{\boldsymbol{x}\sim F}\left[||\boldsymbol{A}^T\boldsymbol{x}||^2\right] \tag{3}$$

We remove the StopGrad operation and set the prediction head (i.e. $\boldsymbol{A}_p$) to $\boldsymbol{I}$. We also trade the normalization term with a regularization term weighted by ξ. The attractive term in SimSiam behaves similar to InfoNCE loss, while the regularization term penalizes the norm of the projected points. The loss scales linearly with $\|\boldsymbol{A}\|^2$, therefore, we impose a norm constraint when optimizing for $\boldsymbol{A}$.

We have already established that maximizing total variance *on its own* in the projected space is not the preferable strategy for dimensionality reduction. In the presence of augmentation (and negatives), self-supervised objectives could take a finer-grained approach by maximizing inter-component variance and keeping intra-component variance low, simultaneously. Next, we will formalize our claim that self-supervised learning is a good proxy for fully supervised techniques (e.g., LDA).

4.2 Main theorem

Nex, we prove that InfoNCE and (simplified) Simsiam can find the Fisher-optimal projection matrix for the class of SharedGMMs.

Theorem 4.1. *Suppose F is a SharedGMM parameterized by $\{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}\}_{k\in[K]}$ and $\hat{F}$ is its Augmentation-enabled Distribution with bias δ. Let S_F be the Fisher subspace (Eqn 1) of F. Denote $\lambda_{\min}$ as the minimum non-zero eigenvalue of $\boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\sum_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T\right)\boldsymbol{\Sigma}^{-\frac{1}{2}}$ and*

$$\boldsymbol{A}_{\text{Info}} = \operatorname*{argmin}_{\boldsymbol{A}\in\mathbb{R}^{d\times r}} \mathcal{L}_{\text{Info}}(\boldsymbol{A}), \qquad \boldsymbol{A}_{\text{Siam}} = \operatorname*{argmin}_{\boldsymbol{A}\in\mathbb{R}^{d\times r},\ ||\boldsymbol{A}||_2\le 1} \mathcal{L}_{\text{Siam}}(\boldsymbol{A})$$

Then for some $r \ge K$, $S_{\text{Info}} \triangleq \text{Col}(\boldsymbol{A}_{\text{Info}})$, $S_{\text{Siam}} \triangleq \text{Col}(\boldsymbol{A}_{\text{Siam}})$, we have :

- $S_{\text{Info}} \subseteq S_F$. *If $\delta = 1$, then $S_{\text{Info}} = S_F$*
- $S_{\text{Siam}} \subseteq S_F$. *If $0 < \xi < \frac{\delta\lambda_{\min}}{1+\lambda_{\min}}$, then $S_{\text{Siam}} = S_F$*

Remark 4.2. *Access to Augmentation-enabled Distribution is **provably sufficient** for learning the Fisher subspace for SharedGMMs matching the performance of supervised LDA. The result implies that class labels which are typically assumed by supervised LDR methods like LDA, are not necessary.*

The proof of Theorem 4.1 can be found in Appendices C and D. For population setting, the fact that self-supervised methods learn mapping onto a subset of the Fisher space implies that the projected points do not capture any *noise* and only contain (a subset of) the useful *features*. Note that spectral methods do not have guarantees of this form. The theorem also states if we have perfect augmentations for InfoNCE loss, i.e. $\delta = 1$, we cover *all directions* in the Fisher space and learn it exactly. The equivalence is true for SimSiam loss when the regularization coefficient ξ is upper bounded appropriately. On a related front, this precludes learning of the collapsing solution for SimSiam loss, verifying the claims in prior work on a simplified version of the loss function. Most importantly, self-supervised methods in question learn *the same subspace* as *supervised* dimensionality reduction methods like LDA which needs the knowledge of underlying components for all the samples.

Technical difficulties. Although we do not provide a proof for $S_{\text{Info}} = S_F$ when $0 < \delta < 1$, we conjecture that it is true, for which we include an empirical discussion in Section 6. Figure 4a suggests that $S_{\text{Info}} = S_F$ even for $\delta < 1$, analysis of which we leave as future work. Additionally, we do not have an exact characterization of InfoNCE solution; we characterize just the column space of $\boldsymbol{A}_{\text{Info}}$. Our empirical results show that the mapped points are well-separated compared to a projection onto the Fisher subspace (see Fig 2d). Furthermore, while we prove that both contrastive and non-contrastive objectives learn the same subspace, further investigation is required for their direct comparison. In Appendix F, we provide an example where $\boldsymbol{A}_{\text{Info}}$ is strictly better than $\boldsymbol{A}_{\text{SimSiam}}$ (for $r < K$). We also present an empirical study on the effect of r (see Figure 4c).

5 Multi-modal Gaussian mixture models

Previously, we considered a setup where a point and its augmentation follow the same distribution. It is a natural assumption for methods like SimCLR and SimSiam where augmentations of points (i.e. images) are defined by transformations (e.g., cropping, color jittering) on the image. This assumption may not hold in general. Text embeddings models like DPR [18] or image-text embedding models like CLIP [20] consider input samples as a pair of points following (possibly) different distribution.

For instance, each sample could be a pair of an image with its corresponding caption (for CLIP) or a search query with a relevant document that answers the query (for DPR). The objective in this *multi-modal* setup is to learn a joint representation space for both modalities. These joint representations can be used downstream for finding similarities between data points belonging to different ambient spaces, and are surprisingly competitive with fully supervised representations [20]. Our goal is to theoretically analyze the representations learned by such model, specifically the CLIP model. Next, we define CLIP-GMM to capture multi-modal data in a theory-friendly setting.

Definition 5.1. (CLIP-GMM) A *CLIP Gaussian mixture* (CLIP-GMM) is defined as the probability distribution $F_{\text{clip}} = \sum_{k\in[K]} w_k \mathcal{N}(\boldsymbol{\mu}_{V,k}, \boldsymbol{\Sigma}_V) \times \mathcal{N}(\boldsymbol{\mu}_{T,k}, \boldsymbol{\Sigma}_T)$ where w_k are the mixture weights, $\{\boldsymbol{\mu}_{V,k}\}_{k\in[K]}, \boldsymbol{\Sigma}_V$ and $\{\boldsymbol{\mu}_{T,k}\}_{k\in[K]}, \boldsymbol{\Sigma}_T$ are the parameters for the two coordinate spaces.

CLIP-GMM is a mixture of product distribution over Gaussians. To elaborate, sampling a pair $(\boldsymbol{x}_V, \boldsymbol{x}_T) \sim F_{\text{clip}}$ is a two step process. We first sample an underlying component index, and then draw independent samples from the component *with the same index* in the respective coordinate space. F_{clip} can be used to define marginal distribution over each coordinate space, for instance, $F_V = \sum_{k\in[K]} w_k \mathcal{N}(\boldsymbol{\mu}_{V,k}, \boldsymbol{\Sigma}_V)$.

5.1 Optimization objective

Following CLIP [20], we want to learn different representation functions for each modality. We let these functions be linear mappings, i.e., $\boldsymbol{A}_V \in \mathbb{R}^{d_1\times r}, \boldsymbol{A}_T \in \mathbb{R}^{d_2\times r}$. Recall that optimal mapping matrices for each coordinate space would have their column space equal to that of Fisher subspace for the marginal distributions F_V, F_T. Concretely, let the representation of a sample $(\boldsymbol{x}_V, \boldsymbol{x}_T)$ be given by $(\boldsymbol{A}_V^T\boldsymbol{x}_V, \boldsymbol{A}_T^T\boldsymbol{x}_T)$, where $\boldsymbol{A}_V, \boldsymbol{A}_T$ are the mappings. Then, we define CLIP InfoNCE loss [20] as,

$$\mathcal{L}_{\text{clip}} = \underset{(\boldsymbol{x}_V,\boldsymbol{x}_T)\sim F_{\text{clip}}}{-\mathbb{E}} \left[(\boldsymbol{A}_T^T\boldsymbol{x}_T)^T \boldsymbol{A}_V^T\boldsymbol{x}_V\right] + \underset{\boldsymbol{x}_T\sim F_T}{\mathbb{E}} \left[\log\left(\underset{\boldsymbol{x}_V\sim F_V}{\mathbb{E}} \left[\exp\left((\boldsymbol{A}_T^T\boldsymbol{x}_T)^T \boldsymbol{A}_V^T\boldsymbol{x}_V\right)\right]\right)\right]$$

Each coordinate space serves as augmentation for the other. Similar to InfoNCE, CLIP InfoNCE attracts embeddings of $\boldsymbol{x}_V$ and $\boldsymbol{x}_T$ via the first term, while regularizing with the Log-Sum-Exp term.

5.2 Results

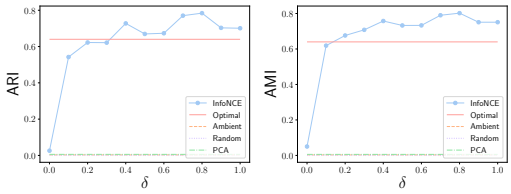
Our main result for CLIP-GMM states that we can learn a subset of Fisher subspace for the constituent modalities by minimizing the CLIP InfoNCE loss. We state our results in the following theorem.

Theorem 5.2. *Suppose $\{w_k, \boldsymbol{\mu}_{V,k}, \boldsymbol{\mu}_{T,k}, \boldsymbol{\Sigma}_V, \boldsymbol{\Sigma}_T\}_{k\in[K]}$ is a CLIP-GMM. Let the Fisher subspace of F_V be $S_{V,F}$ and F_T be $S_{T,F}$. Denote $\boldsymbol{A}_V^*, \boldsymbol{A}_T^*$ as the optimal solution of the CLIP InfoNCE loss,*

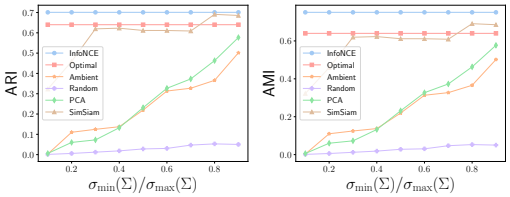
$$\boldsymbol{A}_V^*, \boldsymbol{A}_T^* = \underset{\boldsymbol{A}_V\in\mathbb{R}^{d_1\times r},\ \boldsymbol{A}_T\in\mathbb{R}^{d_2\times r}}{\operatorname{argmin}} \mathcal{L}_{\text{clip}}(\boldsymbol{A}_V, \boldsymbol{A}_T)$$

For any $r \geq K$, let $S_{V,\text{clip}} = \text{Col}(\boldsymbol{A}_V^*)$ and $S_{T,\text{clip}} = \text{Col}(\boldsymbol{A}_T^*)$. Then, $S_{V,\text{clip}} \subseteq S_{V,F}$ and $S_{T,\text{clip}} \subseteq S_{T,F}$.

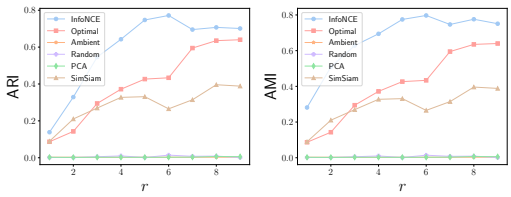
The proof of Theorem 5.2 can be found in Appendix E. In simple words, CLIP InfoNCE learns the subset of the Fisher subspace instead of the *exact space*, which is *weaker* than the single-modal setting. This is due to the fact that the means in the two spaces can vary arbitrarily. Hence, one can choose means and covariances adversarially such that particular directions in the Fisher subspace of either F_V or F_T does not contribute to the inter-component distance and are not learned by the CLIP InfoNCE. For certain special configuration of model parameters, such as $\boldsymbol{\Sigma}_T^{-\frac{1}{2}}\boldsymbol{\mu}_{T,k} = \boldsymbol{\Sigma}_V^{-\frac{1}{2}}\boldsymbol{\mu}_{V,k}$, we can achieve $S_{T,\text{clip}} = S_{T,F}$ and $S_{V,\text{clip}} = S_{V,F}$. The result still shows that self-supervised learning for multi-modal data is better than non-supervised methods. Accessing augmentations are strictly weaker than the knowledge of the labels, and they also naturally occur in the multi-modal setting. The fact that contrastive losses learn a subset of the optimal subspace verifies their capabilities.



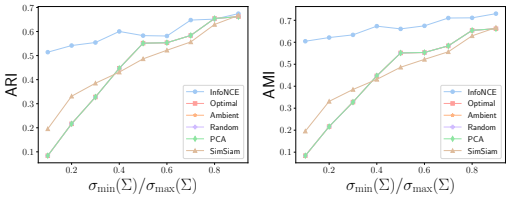
(a) Noise in augmentation



(b) Variance $\perp S_F$



(c) Dimension of projected space



(d) Variance within S_F

Figure 2: We empirically validate our theoretical findings for four main settings. (a) InfoNCE is robust to noise in augmentations for various values of δ. (b) InfoNCE (and SimSiam) are invariant to variance *orthogonal* to Fisher subspace. (c) InfoNCE outperforms spectral methods for every mapping dimension. (d) InfoNCE learns a good scaling within the Fisher subspace

6 Experiments

We validate our theoretical findings with experiments on synthetic and real data. For the synthetic data, we study the effect of noise δ in Augmentation-enabled Distribution , rank r of the projection matrix and condition number of covariance matrix on learned representations. For the real data experiments, we compare self-supervised methods against baselines for clustering CIFAR100 dataset on low dimensional subspaces using K-Means.

6.1 Setup

We evaluate different linear dimensionality reduction methods for SharedGMMs. We adopt a two-step process; the first step is dimensionality reduction with the target method, and the second step is clustering with an out-of-the-box clustering algorithm (i.e. K-Means). We compare the methods based on their clustering performance in the second step using well-known metrics.

Metrics. Following prior work [14] we use Adjusted Rank Index (*ARI*) and Adjusted Mutual Information (*AMI*) to measure the quality of a clustering algorithms. We give the ground truth number of clusters to K-Means as input. ARI and AMI both vary between 0 and 1, with 0 corresponding to random clustering and 1 corresponding to perfect clustering.

Data generation. For synthetic experiments we consider $K = 10$ with equally likely component mixtures. For the CIFAR100 experiments, we randomly sample $K = 20$ classes out of 100. See Suppl G.1 for details.

Methods. We consider learning a mapping matrix $\boldsymbol{A}$ using gradient descent with respect to InfoNCE and SimSiam loss and we compare them againts 5 baselines. Ambient is clustering in ambient dimension. Random projects onto a random r-dimensional subspace. Optimal projects onto top r directions in Fisher subspace. PCA projects onto r-dimensional S_{SVD}. LDA finds a projection that maximizes inter-class variance with respect to intra-class variance (Optimal in the synthetic case).

6.2 Synthetic experiments

We observe that InfoNCE often performs better than Optimal, projection onto the Fisher subspace. We interpret that InfoNCE loss learns to *scale* within the subspace leading to better clustering performance (see App G.2 for orthonormalized plots of InfoNCE/SimSiam). Moreover, SimSiam shows a sub-par performance compared to InfoNCE. We believe this is due the difficulty in optimizing the SimSiam objective (projected gradient descent) compared to InfoNCE.

Table 1: **Clustering performance of linear dimensionality reduction methods on CIFAR100: Bolded** and underlined values indicate the best and second-best scores for each column. We report clustering performance for K-Means on $r = 5, 10, 15, 19$-dimensional subspaces using the linear mappings learned by 6 different methods. We measure the clustering performance using ARI and AMI, which is reported as pairs, in the format **ARI | AMI**. InfoNCE and LDA outperform the rest by significant margins in all mapping dimensions. LDA shows the best performance in terms of ARI, while InfoNCE achieves higher AMI values across the board.

	Mapping Dimension				
Method	**5 dim**	**10 dim**	**15 dim**	**19 dim**	**30 dim**
Random	0.01475 | 0.04694	0.01567 | 0.05241	0.01894 | 0.05796	0.02446 | 0.07188	0.02503 | 0.07856
PCA	0.03360 | 0.09254	0.03292 | 0.09509	0.03435 | 0.09427	0.03154 | 0.09269	0.03361 | 0.09562
Ambient	0.03160 | 0.09481	0.03160 | 0.09481	0.03160 | 0.09481	0.03160 | 0.09481	0.03160 | 0.09481
SimSiam	0.02870 | 0.09581	0.02732 | 0.09114	0.03707 | 0.10830	0.02868 | 0.09498	0.03562 | 0.10540
InfoNCE	0.05065 | **0.12730**	0.05138 | **0.12801**	0.05402 | **0.12835**	0.05285 | **0.12947**	0.05182 | **0.12548**
LDA	**0.05067** | 0.12360	**0.05489** | 0.11782	**0.06352** | 0.12359	**0.05970** | 0.11933	**0.05970** | 0.11933

Effect of noise in augmentations. We vary the noise parameter δ for the AeD with everything else held constant. We can see in Fig 4a that InfoNCE is robust to the variation in δ.

Effect of variance orthogonal to S_F. Increasing the variance *orthogonal to the Fisher subspace* is equivalent to making the pancakes in Fig 1 flatter. We quantify "flatness" as $\sigma_{\min}(\mathbf{\Sigma})/\sigma_{\max}(\mathbf{\Sigma})$. Fig 4b shows that InfoNCE is (almost) invariant to "flatness", while PCA and Ambient fail when the variance orthogonal to Fisher subspace is large (i.e. $\sigma_{\min}(\Sigma)/\sigma_{\max}(\Sigma)$ is small).

Effect of projection dimension. We vary the dimension of the lower dimensional space that we aim to learn. InfoNCE's performance increases with increasing r and finally plateaus.

Effect of variance within S_F. We show that InfoNCE learns a "good" scaling within the Fisher subspace. We set the dimension of ambient space equal to the dimension of mean subspace (i.e. $d = K - 1$). We select $K/2$ orthogonal directions and increase the variance in the subspace *spanned by these direction* by a factor of $\sigma_{\max}(\Sigma)/\sigma_{\min}(\Sigma)$. Since none of the baselines learn scaling in the subspace, they perform almost the same, while InfoNCE outperforms all the baselines.

6.3 Real-data experiments on CIFAR-100

We consider images from CIFAR-100 as inputs instead of generating data from a synthetic GMM. We consider 20 fine-labeled classes (10K images) where each class corresponds to a single component. We convert images into a 256-dimensional vector by subsampling, grayscaling, mean scaling. Images belonging to the same class will be the augmentations of each other. We still consider a linear model and hence have the same baselines. We measure the clustering performance using ARI and AMI on mapping to $r = 5, 10, 15, 19$-dimensional subspaces in Table 1. Surprisingly, InfoNCE could still outperform LDA in certain metrics even in the real data setting where naturally, our data distribution assumptions and augmentation will no longer hold. LDA has the best performance for all dimensions in terms of ARI (first value in each column) which indicates that it excels at maintaining local groupings. On the other hand, InfoNCE achieves larger AMI values underlining that contrastive learning could be better at preserving overall class distributions.

6.4 Real-data experiments on ImageNet

We consider images from ImageNet as inputs instead of generating data from a synthetic GMM. We consider 20 fine-labeled classes (10K images) where each class corresponds to a single component, and images belonging to the same class serve as the augmentations. Since image classification is highly non-linear, we construct a setup to map them to a linearly seperable space. For this we represent each image using ResNet-50's last layer representations (i..e, 2048 dimensional vectors). We measure the clustering performance using ARI and AMI on mapping to $r = 5, 10, 15, 19$-dimensional subspaces in Table 2. We can see that InfoNCE consistently outperforms LDA in this setting.

Table 2: **Clustering performance of linear dimensionality reduction methods on ImageNet: Bolded** and underlined values indicate the best and second-best scores for each column. We report clustering performance for K-Means on $r = 5, 10, 15, 19$-dimensional subspaces using the linear mappings learned by 6 different methods. We measure the clustering performance using ARI and AMI, which is reported as pairs, in the format **ARI | AMI**. InfoNCE and LDA outperform the rest by significant margins in all mapping dimensions. LDA shows the best performance in terms of ARI, while InfoNCE achieves higher AMI values across the board.

	Mapping Dimension				
Method	**5 dim**	**10 dim**	**15 dim**	**19 dim**	**30 dim**
Random	0.03026 | 0.08296	0.07231 | 0.14005	0.13140 | 0.23209	0.14405 | 0.23363	0.26177 | 0.38685
PCA	0.23408 | 0.46447	0.39843 | 0.59117	0.49903 | 0.65898	0.50954 | 0.66159	0.56355 | 0.70163
Ambient	0.48641 | 0.67233	0.48641 | 0.67233	0.48641 | 0.67233	0.48641 | 0.67233	0.48641 | 0.67234
SimSiam	0.37581 | 0.58233	0.56084 | 0.70246	0.58705 | 0.72227	0.60259 | 0.73159	0.62970 | 0.74476
InfoNCE	**0.84451 | 0.88831**	**0.98182 | 0.98115**	**0.92963 | 0.97390**	**0.99621 | 0.99579**	0.93317 | **0.97721**
LDA	0.47112 | 0.69828	0.66967 | 0.83895	0.82550 | 0.89103	0.94744 | 0.94990	**0.94745** | 0.94990

7 Conclusion and limitations

We study contrastive learning in the classical setting of linear dimensionality reduction for GMMs for which we define a generalized notion of imperfect augmentations. Our main result underlines that the contrastive methods learn the Fisher-optimal subspace for the class of shared-covariance GMMs; going beyond the capabilities of unsupervised methods and matching that of fully supervised strategies. Our work particularly focuses on linear mappings, and it is an important open problem to theoretically verify performance of contrastive methods when projectors are non-linear, which is usually the case in practice. Similarly, we acknowledge that our results are not immediately generalizable for data distributions beyond GMMs, which we will investigate as future work.

References

[1] Dimitris Achlioptas and Frank McSherry. On spectral learning of mixtures of distributions. In *International Conference on Computational Learning Theory*, pages 458–469. Springer, 2005.

[2] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

[3] Alice Bizeul, Bernhard Schölkopf, and Carl Allen. A probabilistic model behind self-supervised learning. *arXiv preprint arXiv:2402.01399*, 2024.

[4] S Charles Brubaker and Santosh S Vempala. Isotropic pca and affine-invariant clustering. *Building Bridges: Between Mathematics and Computer Science*, pages 241–281, 2008.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/chen20j.html`.

[7] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[9] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013.

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[11] Jeff Z HaoChen and Tengyu Ma. A theoretical study of inductive biases in contrastive learning. *arXiv preprint arXiv:2211.14699*, 2022.

[12] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021.

[13] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

[14] Heinrich Jiang, Jennifer Jang, and Jakub Lacki. Faster dbscan via subsampled similarity queries. *Advances in Neural Information Processing Systems*, 33:22407–22419, 2020.

[15] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[16] Daniel D Johnson, Ayoub El Hanchi, and Chris J Maddison. Contrastive learning can find an optimal basis for approximately view-invariant functions. *arXiv preprint arXiv:2210.01883*, 2022.

[17] Ravindran Kannan, Santosh Vempala, et al. Spectral algorithms. *Foundations and Trends® in Theoretical Computer Science*, 4(3–4):157–288, 2009.

[18] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[19] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[21] Arora Sanjeev and Ravi Kannan. Learning mixtures of arbitrary gaussians. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, page 247–257, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581133499. doi: 10.1145/380752.380808. URL `https://doi.org/10.1145/380752.380808`.

[22] Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding contrastive learning requires incorporating inductive biases. In *International Conference on Machine Learning*, pages 19250–19286. PMLR, 2022.

[23] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278. PMLR, 2021.

[24] Santosh Vempala and Grant Wang. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860, 2004.

[25] Xiao Wang, Haoqi Fan, Yuandong Tian, Daisuke Kihara, and Xinlei Chen. On the importance of asymmetry for siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16570–16579, 2022.

[26] Zixin Wen and Yuanzhi Li. The mechanism of prediction head in non-contrastive self-supervised learning. *Advances in Neural Information Processing Systems*, 35:24794–24809, 2022.

[27] Runtian Zhai, Bingbin Liu, Andrej Risteski, Zico Kolter, and Pradeep Ravikumar. Understanding augmentation-based self-supervised representation learning via rkhs approximation and regression. *arXiv preprint arXiv:2306.00788*, 2023.

Appendix

A Additional Lemmas

Lemma A.1. *The function given by* $f(\{\boldsymbol{x}_i\}; \boldsymbol{c}) = \log(\sum_i \exp(\boldsymbol{c}^T \boldsymbol{x}_i))$ *where* $\{\boldsymbol{x}_i\}, \boldsymbol{c} \in \mathbb{R}^d$ *is strictly convex in* $\boldsymbol{c} \in \mathrm{Span}(\{\boldsymbol{x}_i\})$.

Proof. Using the definition of strict convexity we need to prove that:

$$\begin{aligned} &f(\{\boldsymbol{x}_i\}; (\lambda \boldsymbol{c}_1 + (1-\lambda)\boldsymbol{c}_2)) \\ <&\lambda f(\{\boldsymbol{x}_i\}; \boldsymbol{c}_1) + (1-\lambda) f(\{\boldsymbol{x}_i\}; \boldsymbol{c}_2) \end{aligned}$$

when $\boldsymbol{c}_1 \neq \boldsymbol{c}_2$. Taking $\exp(.)$ on both sides :

$$\begin{aligned} &\exp(f(\{\boldsymbol{x}_i\}; (\lambda \boldsymbol{c}_1 + (1-\lambda)\boldsymbol{c}_2))) \\ <&\exp(\lambda f(\{\boldsymbol{x}_i\}; \boldsymbol{c}_1)) \exp((1-\lambda) f(\{\boldsymbol{x}_i\}; \boldsymbol{c}_2)) \end{aligned}$$

Simplifying the LHS we have :

$$\begin{aligned} &\exp(f(\{\boldsymbol{x}_i\}; (\lambda \boldsymbol{c}_1 + (1-\lambda)\boldsymbol{c}_2))) \\ =&\sum_i \exp((\lambda \boldsymbol{c}_1 + (1-\lambda)\boldsymbol{c}_2)^T \boldsymbol{x}_i) \\ =&\sum_i \exp(\lambda \boldsymbol{c}_1^T \boldsymbol{x}_i) \exp((1-\lambda)\boldsymbol{c}_2^T \boldsymbol{x}_i) \end{aligned}$$

Using Holders inequality ($\sum_i x_i y_i \leq (\sum_i |x_i|^p)^{\frac{1}{p}} (\sum_i |y_i|^q)^{\frac{1}{q}}$, where $\frac{1}{p} + \frac{1}{q} = 1$) with $p = \frac{1}{\lambda}$ and $q = \frac{1}{1-\lambda}$, we have :

$$\begin{aligned} &\sum_i \exp(\lambda \boldsymbol{c}_1^T \boldsymbol{x}_i) \exp((1-\lambda)\boldsymbol{c}_2^T \boldsymbol{x}_i) \\ \leq&(\sum_i \exp(\frac{\lambda \boldsymbol{c}_1^T \boldsymbol{x}_i}{\lambda}))^{\lambda} (\sum_i \exp(\frac{(1-\lambda)\boldsymbol{c}_2^T \boldsymbol{x}_i}{1-\lambda}))^{(1-\lambda)} \\ =&(\sum_i \exp(\boldsymbol{c}_1^T \boldsymbol{x}_i))^{\lambda} (\sum_i \exp(\boldsymbol{c}_2^T \boldsymbol{x}_i))^{(1-\lambda)} \\ =&\exp(\lambda f(\{\boldsymbol{x}_i\}; \boldsymbol{c}_1)) \exp((1-\lambda) f(\{\boldsymbol{x}_i\}; \boldsymbol{c}_2)) \end{aligned}$$

Note that Holders equality holds only when $\boldsymbol{c}_1^T \boldsymbol{x}_i = \boldsymbol{c}_2^T \boldsymbol{x}_i$ for all i. For $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathrm{Span}\{\boldsymbol{x}_i\}$, this equality holds only when $\boldsymbol{c}_1 = \boldsymbol{c}_2$. Hence, for $\boldsymbol{c}_1 \neq \boldsymbol{c}_2$ we have strict inequality i.e.

$$\begin{aligned} &\sum_i \exp(\lambda \boldsymbol{c}_1^T \boldsymbol{x}_i) \exp((1-\lambda)\boldsymbol{c}_2^T \boldsymbol{x}_i) \\ <&(\sum_i \exp(\frac{\lambda \boldsymbol{c}_1^T \boldsymbol{x}_i}{\lambda}))^{\lambda} (\sum_i \exp(\frac{(1-\lambda)\boldsymbol{c}_2^T \boldsymbol{x}_i}{1-\lambda}))^{(1-\lambda)} \\ =&\exp(\lambda f(\{\boldsymbol{x}_i\}; \boldsymbol{c}_1)) \exp((1-\lambda) f(\{\boldsymbol{x}_i\}; \boldsymbol{c}_2)) \end{aligned}$$

This gives strict convexity. Hence proved. □

B Main Proposition

In this section, we state and prove a key proposition which is used to prove our main result (Thm 4.1)

Proposition B.1. *Suppose F parameterized by $\{w_k, \boldsymbol{\mu}_k, I\}_{k\in[K]}$ be a spherical gaussian mixture model and $\hat{F}$ be its augmentation-enabled gaussian mixture with bias δ (Def 2.2). Let S_F be the fisher subspace (Eqn 1) of F and $\boldsymbol{A}^*$ be the optimal solution of the InfoNCE loss (Eqn 2) :*

$$\boldsymbol{A}^* = \underset{\boldsymbol{A}\in\mathbb{R}^{d\times r}}{\operatorname{argmin}}\, \mathcal{L}(\boldsymbol{A})$$

Then given $r \geq K$, $\mathrm{Col}(\boldsymbol{A}^*) \subseteq S_F$. Moreover if $\delta = 1$, then $\mathrm{Col}(\boldsymbol{A}^*) = S_F$.

We prove the proposition in two parts. In the first part we prove that $\mathrm{Col}(\boldsymbol{A}^*) \subseteq S_F$ for any $\delta > 0$. In the second part we prove that if $\delta = 1$, then $\mathrm{Col}(\boldsymbol{A}^*) = S_F$.

B.1 Column space of A is a subset of Fischer Subspace

We prove that : $\mathrm{Col}(\boldsymbol{A}^*) \subseteq S_F$ when $\delta > 0$

Proof.

$$(\boldsymbol{A}^T\boldsymbol{x})^T(\boldsymbol{A}^T\boldsymbol{y}) = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{y} = \boldsymbol{x}^T\boldsymbol{B}\boldsymbol{y},$$

where $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{A}^T$, i.e. $\boldsymbol{B}$ is positive semi-definite (PSD) matrix of rank r. We substitute into InfoNCE loss to get :

$$\mathcal{L} = -\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}}[\boldsymbol{x}^T\boldsymbol{B}\hat{\boldsymbol{x}}] + \mathbb{E}_{\boldsymbol{x}}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})])]$$

We relax the rank-constraint on $\boldsymbol{B}$ throughout the proof. We show that rank of our optimal solution $\boldsymbol{B}^* \leq K$ which satisfies the rank constraint implicitly (as $K \leq r$).

Note : The above loss function is strictly convex in $\boldsymbol{B}$ (using Lemma A.1) and the minimization of is over a convex set (i.e. set of $\boldsymbol{B} \in \mathbb{S}_+^d$).

Let $\boldsymbol{B}^*$ be the optimal solution. Denote the eigendecomposition of $\boldsymbol{B}^*$ as $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, where $\boldsymbol{\Lambda} \succeq 0$ (as $\boldsymbol{B} \succeq 0$) and $\boldsymbol{U}$ is a unitary matrix. Equivalently :

$$\boldsymbol{B}^* = \sum_i \lambda_i \boldsymbol{u}_i\boldsymbol{u}_i^T \tag{4}$$

where $\boldsymbol{u}_i$ are columns of $\boldsymbol{U}$. Consider the indices where the eigenvalue $\lambda_i > 0$ as $\mathcal{I}$. The solution $\boldsymbol{A}^*$ is hence $\left[\sqrt{\lambda_i}\boldsymbol{u}_i\right]_{i\in\mathcal{I}}$. We aim to show that $\mathrm{Col}(\boldsymbol{A}^*) = \mathrm{Span}\{\boldsymbol{u}_i\}_{i\in\mathcal{I}} \subseteq \mathrm{Span}\{\boldsymbol{\mu}_k\}_{k\in[K]}$.

The condition $\mathrm{Span}\{\boldsymbol{u}_i\}_{i\in\mathcal{I}} \subseteq \mathrm{Span}\{\boldsymbol{\mu}_k\}_{k\in[K]}$ implicitly implies that rank of $\boldsymbol{B}^* \leq K$. This follows because there can't be more than K orthogonal vectors (i.e. columns of U) in a subspace of dimension K (as there are only K means spanning the subspace).

Suppose the condition is not true. Then there exists a unit vector $\boldsymbol{v}$, such that $\boldsymbol{v}^T\boldsymbol{\mu}_k = 0$ for all $k \in [K]$ and $\boldsymbol{v}^T\boldsymbol{u}_i \neq 0$ for some i where $\lambda_i > 0$.

We construct a new matrix $\bar{\boldsymbol{U}}$, whose columns are reflection of $\boldsymbol{U}$ through the plane with normal vector $\boldsymbol{v}$ given by $\boldsymbol{R} = \boldsymbol{I} - 2\boldsymbol{v}\boldsymbol{v}^T$. The reflection matrix is defined such that $\boldsymbol{R}\boldsymbol{\mu}_k = \boldsymbol{R}^T\boldsymbol{\mu}_k = \boldsymbol{\mu}_k$. Define $\bar{\boldsymbol{B}}$ from the constructed $\bar{\boldsymbol{U}}$.

$$\bar{\boldsymbol{U}} = \boldsymbol{R}\boldsymbol{U} = (\boldsymbol{I} - 2\boldsymbol{v}\boldsymbol{v}^T)\boldsymbol{U}$$
$$\bar{\boldsymbol{B}} = \bar{\boldsymbol{U}}\boldsymbol{\Lambda}\bar{\boldsymbol{U}}^T = \boldsymbol{R}\boldsymbol{B}^*\boldsymbol{R}^T$$

$\bar{\boldsymbol{B}} \neq \boldsymbol{B}$. $\bar{\boldsymbol{U}} \neq \boldsymbol{U}$ is still a unitary matrix (product of unitary matrices), is identical to $\boldsymbol{U}$ in $\mathrm{Span}\{\boldsymbol{\mu}_k\}_{k\in[K]}$

$$\bar{\boldsymbol{U}}^T\bar{\boldsymbol{U}} = \boldsymbol{U}^T\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{U} = \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$$
$$\bar{\boldsymbol{U}}^T\boldsymbol{\mu}_k = \boldsymbol{U}^T\boldsymbol{R}^T\boldsymbol{\mu}_k = \boldsymbol{U}^T\boldsymbol{\mu}_k$$
$$\bar{\boldsymbol{U}}\boldsymbol{\mu}_k = \boldsymbol{U}\boldsymbol{R}\boldsymbol{\mu}_k = \boldsymbol{U}\boldsymbol{\mu}_k$$

The first term in the loss $\mathcal{L}$ at $\boldsymbol{B}^*$ can be simplified as

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}}[\boldsymbol{x}^T\boldsymbol{B}^*\hat{\boldsymbol{x}}] \\
=&\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}}[\langle \boldsymbol{x}\hat{\boldsymbol{x}}^T, \boldsymbol{B}^*\rangle] \\
=&\langle \mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}}[\boldsymbol{x}\hat{\boldsymbol{x}}^T], \boldsymbol{B}^*\rangle \\
=&\langle \sum_k w_k\boldsymbol{\mu}_k(\delta\boldsymbol{\mu}_k + (1-\delta)(\sum_j w_j\boldsymbol{\mu}_j))^T, \boldsymbol{B}^*\rangle \\
=&\langle \delta\sum_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T + (1-\delta)(\sum_k w_k\boldsymbol{\mu}_k)(\sum_j w_j\boldsymbol{\mu}_j)^T, \boldsymbol{B}^*\rangle
\end{aligned}$$

This is where we use the fact that the augmentation is generated from the augmentation oracle.

We can now see that

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}}[\boldsymbol{x}^T\bar{\boldsymbol{B}}\hat{\boldsymbol{x}}] \\
=&\langle \delta\sum_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T + (1-\delta)(\sum_k w_k\boldsymbol{\mu}_k)(\sum_j w_j\boldsymbol{\mu}_j)^T, \bar{\boldsymbol{B}}\rangle \\
=&\langle \delta\sum_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T + (1-\delta)(\sum_k w_k\boldsymbol{\mu}_k)(\sum_j w_j\boldsymbol{\mu}_j)^T, \boldsymbol{R}\boldsymbol{B}\boldsymbol{R}^T\rangle \\
=&\langle \delta\sum_k w_k\boldsymbol{R}^T\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T\boldsymbol{R} + \\
&(1-\delta)(\sum_k w_k\boldsymbol{R}^T\boldsymbol{\mu}_k)(\sum_j w_j\boldsymbol{R}^T\boldsymbol{\mu}_j)^T, \boldsymbol{B}^*\rangle \\
=&\langle \delta\sum_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T + (1-\delta)(\sum_k w_k\boldsymbol{\mu}_k)(\sum_j w_j\boldsymbol{\mu}_j)^T, \boldsymbol{B}^*\rangle \\
=&\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}}[\boldsymbol{x}^T\boldsymbol{B}^*\hat{\boldsymbol{x}}]
\end{aligned}$$

Now see analyze the second term in $\mathcal{L}$ where we prove :

$$\mathbb{E}_{\boldsymbol{x}}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\bar{\boldsymbol{B}}\tilde{\boldsymbol{x}})])] = \mathbb{E}_{\boldsymbol{x}}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}^*\tilde{\boldsymbol{x}})])]$$

For this we show that the random variables $\boldsymbol{x}^T\boldsymbol{B}^*\tilde{\boldsymbol{x}}$ and $\boldsymbol{x}^T\bar{\boldsymbol{B}}\tilde{\boldsymbol{x}}$ have identical distribution. We first simplify $\boldsymbol{x}^T\boldsymbol{B}^*\tilde{\boldsymbol{x}}$ below.

$$\begin{aligned}
&\boldsymbol{x}^T\bar{\boldsymbol{B}}\tilde{\boldsymbol{x}} \\
=&\boldsymbol{x}^T\boldsymbol{R}\boldsymbol{B}^*\boldsymbol{R}\tilde{\boldsymbol{x}} \\
=&(\boldsymbol{x}-\boldsymbol{\mu}_z+\boldsymbol{\mu}_z)^T\boldsymbol{R}\boldsymbol{B}^*\boldsymbol{R}(\tilde{\boldsymbol{x}}-\boldsymbol{\mu}_{\tilde{z}}+\boldsymbol{\mu}_{\tilde{z}}) \\
=&((\boldsymbol{x}-\boldsymbol{\mu}_z)^T\boldsymbol{R}+\boldsymbol{\mu}_z^T\boldsymbol{R})\boldsymbol{B}^*(\boldsymbol{R}(\tilde{\boldsymbol{x}}-\boldsymbol{\mu}_{\tilde{z}})+\boldsymbol{R}\boldsymbol{\mu}_{\tilde{z}}) \\
=&((\boldsymbol{x}-\boldsymbol{\mu}_z)^T\boldsymbol{R}+\boldsymbol{\mu}_z^T)\boldsymbol{B}^*(\boldsymbol{R}(\tilde{\boldsymbol{x}}-\boldsymbol{\mu}_{\tilde{z}})+\boldsymbol{\mu}_{\tilde{z}})
\end{aligned}$$

where $\boldsymbol{\mu}_z^T\boldsymbol{R} = \boldsymbol{\mu}_z^T(\boldsymbol{I}-2\boldsymbol{v}\boldsymbol{v}^T) = \boldsymbol{\mu}_z^T - 2(\boldsymbol{\mu}_z^T\boldsymbol{v})\boldsymbol{v}^T = \boldsymbol{\mu}_z^T$. Now we prove that their distributions are identical.

$$
\begin{aligned}
&\Pr(\boldsymbol{x}^T\bar{\boldsymbol{B}}\tilde{\boldsymbol{x}} \le c)\\
=&\Pr(((\boldsymbol{x}-\boldsymbol{\mu}_z)^T\boldsymbol{R}+\boldsymbol{\mu}_z^T)\boldsymbol{B}^*(\boldsymbol{R}(\tilde{\boldsymbol{x}}-\boldsymbol{\mu}_{\tilde{z}})+\boldsymbol{\mu}_{\tilde{z}}) \le c)\\
=&\Pr(((\boldsymbol{x}-\boldsymbol{\mu}_z)^T+\boldsymbol{\mu}_z^T)\boldsymbol{B}^*((\tilde{\boldsymbol{x}}-\boldsymbol{\mu}_{\tilde{z}})+\boldsymbol{\mu}_{\tilde{z}}) \le c)\\
=&\Pr(\boldsymbol{x}\boldsymbol{B}^*\tilde{\boldsymbol{x}} \le c)
\end{aligned}
$$

We use the fact that $(\boldsymbol{x}-\boldsymbol{\mu}_z)^T(\boldsymbol{I}-2\boldsymbol{v}\boldsymbol{v}^T)$ is identically distributed to $(\boldsymbol{x}-\boldsymbol{\mu}_z)$ i.e. $\mathcal{N}(0,\boldsymbol{I})$.

Now since $\boldsymbol{x}^T\boldsymbol{B}^*\tilde{\boldsymbol{x}}, \boldsymbol{x}^T\bar{\boldsymbol{B}}\tilde{\boldsymbol{x}}$ are identically distributed, second term in the loss are also identically distributed and have the same expectation i.e.

$$
\mathbb{E}_{\boldsymbol{x}}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}^*\tilde{\boldsymbol{x}})])] = \mathbb{E}_{\boldsymbol{x}}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\bar{\boldsymbol{B}}\tilde{\boldsymbol{x}})])]
$$

This proves that $\mathcal{L}$ is identical for $\boldsymbol{B}^*$ and $\boldsymbol{B}'$. But since our loss is strictly convex we have $\mathcal{L}(\frac{\boldsymbol{B}^*+\bar{\boldsymbol{B}}}{2}) < \frac{1}{2}(\mathcal{L}(\boldsymbol{B}^*)+\mathcal{L}(\bar{\boldsymbol{B}})) = \mathcal{L}(\boldsymbol{B}^*)$. This contradicts the fact that $\boldsymbol{B}^*$ is optimal in $\mathbb{S}_+^d$. Hence proved. □

B.2 Column space of A is equal to Fischer Subspace

We prove that : $\mathrm{Col}(\boldsymbol{A}^*) = S_F$ when $\delta = 1$

Proof. Consider the eigendecomposition for the optimal solution $\boldsymbol{B}^*$ (Eq 5). Suppose there is a direction $\boldsymbol{v} \in \mathrm{Span}\big(\{\boldsymbol{\mu}_k\}_k\big)$ which is not in $\mathrm{Span}\big(\{\boldsymbol{u}_i\}_{i\in\mathcal{I}}\big)$.

Without loss of generality assume $\boldsymbol{v}^T\boldsymbol{u}_i = 0\ \forall i \in \mathcal{I}$ (if not then project onto the null space and re-normalize). Now since $\lambda_j = 0\ \forall i \notin \mathcal{I}$, we can rotate the eigenvectors $[\boldsymbol{u}_j]_{j\notin\mathcal{I}}$ with a unitary matrix such that $\boldsymbol{u}_j = v$ for some j. This operation doesn't change $\boldsymbol{B}^*$

That implies there exists $\boldsymbol{u}_j$ such that $\boldsymbol{u}_j \in \mathrm{Span}\big(\{\boldsymbol{\mu}_k\}_k\big)$ and $\lambda_j = 0$. We now show that

$$
\frac{\partial\mathcal{L}}{\partial\lambda_j}\Big|_{\boldsymbol{B}=\boldsymbol{B}^*} < 0
$$

Hence $\lambda_j > 0$ for optimal $\boldsymbol{B}^*$.

First consider the derivative of first term of $\mathcal{L}$ with λ_j

$$
\begin{aligned}
&\frac{\partial\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}}[\boldsymbol{x}^T\boldsymbol{B}\hat{\boldsymbol{x}}]}{\partial\lambda_j} = \langle\sum_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T, \frac{\partial\boldsymbol{B}}{\partial\lambda_j}\rangle\\
=&\langle\sum_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T, \boldsymbol{u}_j\boldsymbol{u}_j^T\rangle = \sum_k w_k(\boldsymbol{\mu}_k^T\boldsymbol{u}_j)^2 = \sum_k w_k a_k^2
\end{aligned}
$$

where $a_k = \boldsymbol{\mu}_k^T\boldsymbol{u}_j$. Since $\boldsymbol{u}_j \in \mathrm{Span}\big(\{\boldsymbol{\mu}_k\}_k\big)$ that implies there exists a non-zero a_k. Hence we have that $\frac{\partial\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}}[\boldsymbol{x}^T\boldsymbol{B}\hat{\boldsymbol{x}}]}{\partial\lambda_j} > 0$ for any $\boldsymbol{B} \neq 0$.

For the second term in the loss we have :

$$
\begin{aligned}
&\frac{\partial \mathbb{E}_{\boldsymbol{x}}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})])]}{\partial \lambda_j} \\
=&\mathbb{E}_{\boldsymbol{x}}\left[\frac{\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})\frac{\partial \boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}}}{\partial \lambda_j}]}{\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})]}\right] \\
=&\mathbb{E}_{\boldsymbol{x}}\left[\frac{\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})(\boldsymbol{x}^T\boldsymbol{u}_j)(\tilde{\boldsymbol{x}}^T\boldsymbol{u}_j)]}{\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})]}\right] \\
=&\mathbb{E}_{\boldsymbol{x}}\left[(\boldsymbol{x}^T\boldsymbol{u}_j)\frac{\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})(\tilde{\boldsymbol{x}}^T\boldsymbol{u}_j)]}{\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})]}\right] \\
=&\mathbb{E}_{\boldsymbol{x}}\left[(\boldsymbol{x}^T\boldsymbol{u}_j)\mathbb{E}_{\tilde{\boldsymbol{x}}}\left[\frac{\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})}{\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})]}\tilde{\boldsymbol{x}}^T\boldsymbol{u}_j\right]\right] \\
=&\mathbb{E}_{\boldsymbol{x}}\left[(\boldsymbol{x}^T\boldsymbol{u}_j)\mathbb{E}_{\tilde{\boldsymbol{x}}}\left[g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B})\tilde{\boldsymbol{x}}^T\boldsymbol{u}_j\right]\right]
\end{aligned}
$$

where we define $g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B})$ as :

$$
g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B}) \triangleq \frac{\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})}{\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})]}
$$

with $\mathbb{E}_{\tilde{\boldsymbol{x}}}[g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B})] = 1$. Also define $\boldsymbol{x}_\perp = \boldsymbol{u}_j(\boldsymbol{u}_j^T\boldsymbol{x})$ and $\boldsymbol{x}_\parallel = (I - \boldsymbol{u}_j\boldsymbol{u}_j^T)\boldsymbol{x}$. Hence $\boldsymbol{x} = \boldsymbol{x}_\perp + \boldsymbol{x}_\parallel$. Notice that since $\lambda_j = 0$ for $\boldsymbol{B}^*$, we have

$$
g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B}^*) = g(\tilde{\boldsymbol{x}}_\parallel;\boldsymbol{x},\boldsymbol{B}^*) = g(\tilde{\boldsymbol{x}}_\parallel;\boldsymbol{x}_\parallel,\boldsymbol{B}^*)
$$

We evaluate the inner term in the above expression at $\boldsymbol{B} = \boldsymbol{B}^*$:

$$
\begin{aligned}
&\mathbb{E}_{\tilde{\boldsymbol{x}}}\left[g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B}^*)\tilde{\boldsymbol{x}}^T\boldsymbol{u}_j\right] \\
=&\sum_k w_k \mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{N}(\boldsymbol{\mu}_k,I)}\left[g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B}^*)\tilde{\boldsymbol{x}}^T\boldsymbol{u}_j\right] \\
=&\sum_k w_k \mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{N}(\boldsymbol{\mu}_k,I)}\left[g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B}^*)\tilde{\boldsymbol{x}}^T\boldsymbol{u}_j\right]
\end{aligned}
$$

We look at each of the expectations individually

$$
\begin{aligned}
&\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{N}(\boldsymbol{\mu}_k,I)}\left[g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B}^*)\tilde{\boldsymbol{x}}^T\boldsymbol{u}_j\right] \\
=&\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{N}(\boldsymbol{\mu}_k,I)}\left[g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B}^*)(a_k + \tilde{\boldsymbol{z}}^T\boldsymbol{u}_j)\right] \\
=&\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{N}(\boldsymbol{\mu}_k,I)}\left[g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B}^*)a_k\right] + \\
&\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{N}(\boldsymbol{\mu}_k,I)}\left[g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B}^*)(\tilde{\boldsymbol{z}}_\perp^T\boldsymbol{u}_j)\right] \\
=&a_k\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{N}(\boldsymbol{\mu}_k,I)}\left[g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B}^*)\right] = a_k h_k(\boldsymbol{x};\boldsymbol{B}^*)
\end{aligned}
$$

where $\tilde{\boldsymbol{z}}$ is the noise in $\tilde{\boldsymbol{x}}$ and $h_k(\boldsymbol{x}; \boldsymbol{B}^*) = \mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_k, I)} \Big[g(\tilde{\boldsymbol{x}}; \boldsymbol{x}, \boldsymbol{B}^*) \Big]$. We use the property that $\tilde{\boldsymbol{z}}_\perp$ is independent of $g(\tilde{\boldsymbol{x}}; \boldsymbol{x}, \boldsymbol{B}^*) = g(\tilde{\boldsymbol{x}}_\parallel; \boldsymbol{x}, \boldsymbol{B}^*)$ to show that its equal to 0.

$$
\begin{aligned}
&\mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_k, I)} \Big[g(\tilde{\boldsymbol{x}}_\parallel; \boldsymbol{x}, \boldsymbol{B}^*) (\tilde{\boldsymbol{z}}_\perp^T \boldsymbol{u}_j) \Big] \\
=&\mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_k, I)} \Big[g(\tilde{\boldsymbol{x}}_\parallel; \boldsymbol{x}, \boldsymbol{B}^*) \Big] \mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_k, I)} \Big[\tilde{\boldsymbol{z}}_\perp^T \boldsymbol{u}_j \Big] \\
=&\mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_k, I)} \Big[g(\tilde{\boldsymbol{x}}_\parallel; \boldsymbol{x}, \boldsymbol{B}^*) \Big] * 0 = 0
\end{aligned}
$$

Hence we have

$$
\mathbb{E}_{\tilde{\boldsymbol{x}}} \Big[g(\tilde{\boldsymbol{x}}; \boldsymbol{x}, \boldsymbol{B}^*) \tilde{\boldsymbol{x}}^T \boldsymbol{u}_j \Big] = \sum_k w_k a_k h_k(\boldsymbol{x}; \boldsymbol{B}^*)
$$

We have $\sum_k w_k h_k(\boldsymbol{x}) = 1$

$$
\begin{aligned}
&\sum_k w_k h_k(\boldsymbol{x}; \boldsymbol{B}) \\
=&\sum_k w_k \mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_k, I)} \Big[g(\tilde{\boldsymbol{x}}_\parallel; \boldsymbol{x}, \boldsymbol{B}) \Big] \\
=&\sum_k w_k \mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}_k, I)} \left[\frac{\exp(\boldsymbol{x}^T \boldsymbol{B} \tilde{\boldsymbol{x}})}{\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T \boldsymbol{B} \tilde{\boldsymbol{x}})]} \right] \\
=&\mathbb{E}_{\tilde{\boldsymbol{x}}} \left[\frac{\exp(\boldsymbol{x}^T \boldsymbol{B} \tilde{\boldsymbol{x}})}{\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T \boldsymbol{B} \tilde{\boldsymbol{x}})]} \right] = 1
\end{aligned}
$$

While $g(\tilde{\boldsymbol{x}}; \boldsymbol{x}, \boldsymbol{B})$, is the weight $\boldsymbol{x}$ gives to a point $\tilde{\boldsymbol{x}}$, $h_k(\boldsymbol{x}; \boldsymbol{B})$ can be interpreted as a weight $\boldsymbol{x}$ gives to cluster k (i.e. expectation of g over a cluster).

Going back to the expression at $\boldsymbol{B} = \boldsymbol{B}^*$

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}} \Big[(\boldsymbol{x}^T \boldsymbol{u}_j) \mathbb{E}_{\tilde{\boldsymbol{x}}} \Big[g(\tilde{\boldsymbol{x}}; \boldsymbol{x}, \boldsymbol{B}^*) \tilde{\boldsymbol{x}}^T \boldsymbol{u}_j \Big] \Big] \\
=&\mathbb{E}_{\boldsymbol{x}} \Big[(\boldsymbol{x}^T \boldsymbol{u}_j) \sum_k w_k a_k h_k(\boldsymbol{x}; \boldsymbol{B}^*) \Big] \\
=&\sum_k w_k a_k \mathbb{E}_{\boldsymbol{x}} \Big[(\boldsymbol{x}^T \boldsymbol{u}_j) h_k(\boldsymbol{x}; \boldsymbol{B}^*) \Big] \\
=&\sum_k w_k a_k \mathbb{E}_{\boldsymbol{x}} \Big[(\boldsymbol{x}^T \boldsymbol{u}_j) h_k(\boldsymbol{x}; \boldsymbol{B}^*) \Big] \\
=&\sum_k w_k a_k \sum_{k'} w_{k'} \mathbb{E}_{\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_{k'}, I)} \Big[(\boldsymbol{x}^T \boldsymbol{u}_j) h_k(\boldsymbol{x}; \boldsymbol{B}^*) \Big]
\end{aligned}
$$

h inherits the property $h_k(\boldsymbol{x}; \boldsymbol{B}^*) = h_k(\boldsymbol{x}_\parallel; \boldsymbol{B}^*)$ from g (this is because $\lambda_j = 0$ in $\boldsymbol{B}^*$). Evaluating the inner expression we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}_{k'},I)}\left[(\boldsymbol{x}^T\boldsymbol{u}_j)h_k(\boldsymbol{x};\boldsymbol{B}^*)\right] \\
=&\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}_{k'},I)}\left[(a_{k'}+\boldsymbol{z}_\perp^T\boldsymbol{u}_j)h_k(\boldsymbol{x};\boldsymbol{B}^*)\right] \\
=&a_{k'}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}_{k'},I)}\left[h_k(\boldsymbol{x};\boldsymbol{B}^*)\right] \\
+&\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}_{k'},I)}\left[(\boldsymbol{z}_\perp^T\boldsymbol{u}_j)h_k(\boldsymbol{x};\boldsymbol{B}^*)\right] \\
=&a_{k'}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}_{k'},I)}\left[h_k(\boldsymbol{x};\boldsymbol{B}^*)\right] = a_{k'}f_{k',k}
\end{aligned}$$

where $\boldsymbol{z}$ is the noise in $\boldsymbol{x}$. We use the property that $\boldsymbol{z}_\perp$ is independent of $h_k(\boldsymbol{x}; \boldsymbol{B}^*) = h_k(\boldsymbol{x}_\parallel; \boldsymbol{B}^*)$ to show that its equal to 0.

$$\begin{aligned}
\sum_k w_k f_{k',k} =& \sum_k w_k \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}_{k'},I)}\left[h_k(\boldsymbol{x};\boldsymbol{B}^*)\right] \\
&\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}_{k'},I)}\left[\sum_k w_k h_k(\boldsymbol{x};\boldsymbol{B}^*)\right] = 1
\end{aligned}$$

since $\sum_k w_k h_k(\boldsymbol{x}; \boldsymbol{B}^*) = 1$.

Define a matrix $\boldsymbol{F} \in \mathbb{R}^{K\times K}$ as $\boldsymbol{F}_{i,j} = f_{i,j}$ and $\boldsymbol{W} = \operatorname{diag}(w_1, w_2, \ldots, w_K)$ and $\boldsymbol{a} = [a_1, a_2, \ldots, a_K]^T$.

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}}\left[(\boldsymbol{x}^T\boldsymbol{u}_j)\mathbb{E}_{\tilde{\boldsymbol{x}}}\left[g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B}^*)\tilde{\boldsymbol{x}}^T\boldsymbol{u}_j\right]\right] \\
=&\sum_k w_k a_k \sum_{k'} w_{k'}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}_{k'},I)}\left[(\boldsymbol{x}^T\boldsymbol{u}_j)h_k(\boldsymbol{x};\boldsymbol{B}^*)\right] \\
=&\sum_k w_k a_k \sum_{k'} w_{k'} a_{k'} f_{k',k} \\
=&\boldsymbol{a}^T\boldsymbol{W}\boldsymbol{F}\boldsymbol{W}\boldsymbol{a} = (\sqrt{\boldsymbol{W}}\boldsymbol{a})^T\sqrt{\boldsymbol{W}}\boldsymbol{F}\sqrt{\boldsymbol{W}}(\sqrt{\boldsymbol{W}}\boldsymbol{a})
\end{aligned}$$

Eigenvalues of $\sqrt{\boldsymbol{W}}\boldsymbol{F}\sqrt{\boldsymbol{W}}$ are equal to eigenvalues of $\boldsymbol{F}\boldsymbol{W}$ (since this is a similarity transform). And if $\boldsymbol{v}$ is eigenvector of $\boldsymbol{F}\boldsymbol{W}$, then $\sqrt{\boldsymbol{W}}\boldsymbol{v}$ is the eigenvector for $\sqrt{\boldsymbol{W}}\boldsymbol{F}\sqrt{\boldsymbol{W}}$.

Since $\boldsymbol{F}$ has expectation of exponentials as its entries i.e. $\boldsymbol{F}_{i,j} > 0$ and $\boldsymbol{W} > 0$ from definition. Hence we have that every entry of $\boldsymbol{F}\boldsymbol{W}$ is strictly greater than 0. The Perron–Frobenius eigenvalue r is given by

$$\min_i \sum_j (\boldsymbol{F}\boldsymbol{W})_{i,j} \le r \le \max_i \sum_j (\boldsymbol{F}\boldsymbol{W})_{i,j}$$

But we have that for every i the sum is equal to 1.

$$\sum_j (\boldsymbol{F}\boldsymbol{W})_{i,j} = \sum_j \sum_l (\boldsymbol{F}_{i,l}\boldsymbol{W}_{l,j}) = \sum_j f_{i,j} w_j = 1$$

Hence $r = 1$ and the Perron vector is simply $\mathbf{1} = [1, 1, \ldots, 1] \in \mathbb{R}^K$. Hence we show that eigenvector with largest eigenvalue (i.e. 1) for $\sqrt{\boldsymbol{W}}\boldsymbol{F}\sqrt{\boldsymbol{W}}$ is $\sqrt{\boldsymbol{W}}\mathbf{1}$. Hence we have that

$$(\sqrt{\boldsymbol{W}}\boldsymbol{a})^T\sqrt{\boldsymbol{W}}\boldsymbol{F}\sqrt{\boldsymbol{W}}(\sqrt{\boldsymbol{W}}\boldsymbol{a}) \leq ||\sqrt{\boldsymbol{W}}\boldsymbol{a}||^2 = \sum_k w_k a_k^2$$

The equality only holds when $\boldsymbol{a} \in \mathrm{Span}(\mathbf{1})$, i.e. all a_k are equal. This is not true (since $\sum_k a_k = 0$ and there exists a nonzero a_k). Hence

$$\mathbb{E}_{\boldsymbol{x}}\left[(\boldsymbol{x}^T\boldsymbol{u}_j)\mathbb{E}_{\tilde{\boldsymbol{x}}}\left[g(\tilde{\boldsymbol{x}};\boldsymbol{x},\boldsymbol{B}^*)\tilde{\boldsymbol{x}}^T\boldsymbol{u}_j\right]\right] < \sum_k w_k a_k^2$$

The loss for the whole term is

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\lambda_j}\Big|_{\boldsymbol{B}=\boldsymbol{B}^*} &= -\frac{\partial\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}}[\boldsymbol{x}^T\boldsymbol{B}\hat{\boldsymbol{x}}]}{\partial\lambda_j} \\ &\quad + \frac{\partial\mathbb{E}_{\boldsymbol{x}}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})])]}{\partial\lambda_j} \\ &< -\sum_k w_k a_k^2 + \sum_k w_k a_k^2 = 0\end{aligned}$$

Hence proved.

□

C Proof for InfoNCE loss

In this section, we use Proposition B.1 to prove our main theorem. We recommend the reader to first go through Proposition B.1 and it's proof (in Appendix B) to understand the proof for the main theorem.

Proof.

$$\mathcal{L} = -\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}}}[\boldsymbol{x}^T\boldsymbol{B}\hat{\boldsymbol{x}}] + \mathbb{E}_{\boldsymbol{x}}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}}[\exp(\boldsymbol{x}^T\boldsymbol{B}\tilde{\boldsymbol{x}})])]$$

where $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{A}^T$, i.e. $\boldsymbol{B}$ is positive semi-definite (PSD) matrix of rank r. The above step follows from Proposition B.1. We relax the rank-constraint on $\boldsymbol{B}$. We finally show that rank of $\boldsymbol{B}^*$ i.e. the optimal solution $\leq K$ which satisfies the above condition as $K \leq r$.

Now consider a change of variables given by $\boldsymbol{x}' = \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{x}$. This implies that $\boldsymbol{x}'$ now follows a GMM with means given by $\{\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\mu}_k\}_{k\in[K]}$ and isotropic covariance $\boldsymbol{I}$ (as $\mathbb{E}[\boldsymbol{x}'\boldsymbol{x}'^T] = \mathbb{E}[\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{\Sigma}^{-\frac{1}{2}}] = \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{\Sigma}^{-\frac{1}{2}} = \boldsymbol{I}$).

The loss be hence be written as :

$$\begin{aligned}\mathcal{L} = &-\mathbb{E}_{\boldsymbol{x}',\hat{\boldsymbol{x}}'}[\boldsymbol{x}'^T\boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{B}\boldsymbol{\Sigma}^{\frac{1}{2}}\hat{\boldsymbol{x}}'] \\ &+\mathbb{E}_{\boldsymbol{x}'}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}'}[\exp(\boldsymbol{x}'^T\boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{B}\boldsymbol{\Sigma}^{\frac{1}{2}}\tilde{\boldsymbol{x}}')])]\end{aligned}$$

Now let $\boldsymbol{B}' = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{B}\boldsymbol{\Sigma}^{\frac{1}{2}}$. $\boldsymbol{B}'$ is also a PSD matrix. Hence loss can be written as :

$$\mathcal{L} = -\mathbb{E}_{\boldsymbol{x}',\hat{\boldsymbol{x}}'}[\boldsymbol{x}'^T\boldsymbol{B}'\hat{\boldsymbol{x}}'] + \mathbb{E}_{\boldsymbol{x}'}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}'}[\exp(\boldsymbol{x}'^T\boldsymbol{B}'\tilde{\boldsymbol{x}}')])]$$

Let the optimal $\boldsymbol{B}'^*$ be denoted by $\sum_i \lambda_i\boldsymbol{u}_i\boldsymbol{u}_i^T$ and let $\mathcal{I}$ be set of indices with $\lambda_i > 0$ (note $\lambda_i \geq 0\forall i$). But from Proposition B.1, we know that , where $\mathrm{Span}\{\boldsymbol{u}_i\}_{i\in\mathcal{I}} = \mathrm{Span}\{\boldsymbol{\mu}'_k\}_{k\in[K]}$ where $\{\boldsymbol{\mu}'_k\}$ is the set of means.

For an invertible $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}_k' = \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\mu}_k$, we have

$$\begin{aligned}
&\text{Span}\{\boldsymbol{u}_i\}_{i\in\mathcal{I}} = \text{Span}\{\boldsymbol{\mu}_k'\} \\
\implies &\text{Span}\{\boldsymbol{u}_i\}_{i\in\mathcal{I}} = \text{Span}\{\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\mu}_k\} \\
\implies &\text{Span}\{\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{u}_i\}_{i\in\mathcal{I}} = \text{Span}\{\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k\}
\end{aligned}$$

Substituting and $\boldsymbol{B}^* = \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{B}'^*\boldsymbol{\Sigma}^{-\frac{1}{2}}$ we get :

$$\begin{aligned}
\boldsymbol{B}^* &= \boldsymbol{\Sigma}^{-\frac{1}{2}}(\sum_i \lambda_i \boldsymbol{u}_i\boldsymbol{u}_i^T)\boldsymbol{\Sigma}^{-\frac{1}{2}} \\
&= \sum_i (\sqrt{\lambda_i}\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{u}_i)(\sqrt{\lambda_i}\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{u}_i)^T
\end{aligned}$$

Hence the optimal solution in the original space $\boldsymbol{A}^* = \left[\sqrt{\lambda_i}\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{u}_i\right]_{i\in\mathcal{I}}$. We proved that $\text{Span}\{\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{u}_i\}_{i\in\mathcal{I}} = \text{Span}\{\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k\}_{k\in[K]}$. This implies that column space of $\text{Col}(\boldsymbol{A}^*) = \text{Span}\{\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{u}_i\}_{i\in\mathcal{I}} = \text{Span}\{\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k\}_{k\in[K]}$. Hence proved.

□

D Proof for SimSiam Loss

In this section, we use the same methodology as in part of Proposition B.1 to prove the theorem. First we use the strict convexity of the loss to show that the solution lies in the fisher subspace. Afterwards, by differentiating the loss function w.r.t. any direction in the fisher subspace, we show that all directions in the subspace should be learnt assuming sufficient capacity.

Proof. We write the objective by computing the expectations as :

$$\begin{aligned}
\mathcal{L}_{SS}(\boldsymbol{A}) &= -\mathop{\mathbb{E}}_{(\boldsymbol{x},\hat{\boldsymbol{x}})\sim\hat{F}}\left[(\boldsymbol{A}^T\boldsymbol{x})^T\boldsymbol{A}^T\hat{\boldsymbol{x}}\right] + \xi\mathop{\mathbb{E}}_{\boldsymbol{x}\sim F}\left[||\boldsymbol{A}^T\boldsymbol{x}||^2\right] \\
&= -\mathop{\mathbb{E}}_{(\boldsymbol{x},\hat{\boldsymbol{x}})\sim\hat{F}}\left[\langle\boldsymbol{A}\boldsymbol{A}^T, \hat{\boldsymbol{x}}\boldsymbol{x}^T\rangle\right] + \xi\mathop{\mathbb{E}}_{\boldsymbol{x}\sim F}\left[\langle\boldsymbol{A}\boldsymbol{A}^T, \boldsymbol{x}\boldsymbol{x}^T\rangle\right] \\
&= -\langle\boldsymbol{A}\boldsymbol{A}^T, \mathop{\mathbb{E}}_{(\boldsymbol{x},\hat{\boldsymbol{x}})\sim\hat{F}}[\hat{\boldsymbol{x}}\boldsymbol{x}^T]\rangle + \xi\langle\boldsymbol{A}\boldsymbol{A}^T, \mathop{\mathbb{E}}_{\boldsymbol{x}\sim F}[\boldsymbol{x}\boldsymbol{x}^T]\rangle \\
&= -\langle\boldsymbol{A}\boldsymbol{A}^T, \delta\sum_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T\rangle + \xi\langle\boldsymbol{A}\boldsymbol{A}^T, \sum_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T + \boldsymbol{\Sigma}\rangle \\
&= \langle\boldsymbol{B}, (-\delta+\xi)\boldsymbol{M}\rangle + \xi\langle\boldsymbol{B}, \boldsymbol{\Sigma}\rangle \\
&= \langle\boldsymbol{B}, (-\delta+\xi)\boldsymbol{M} + \xi\boldsymbol{\Sigma}\rangle \\
&= \langle\boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{B}\boldsymbol{\Sigma}^{\frac{1}{2}}, (-\delta+\xi)\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{M}\boldsymbol{\Sigma}^{-\frac{1}{2}} + \xi\boldsymbol{I}\rangle \\
&= \langle\boldsymbol{B}', (-\delta+\xi)\boldsymbol{M}' + \xi\boldsymbol{I}\rangle
\end{aligned}$$

where we have $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{A}^T$, $\boldsymbol{M} = \sum_k w_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T$, $\boldsymbol{B}' = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{B}\boldsymbol{\Sigma}^{\frac{1}{2}}$ and $\boldsymbol{M}' = \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{M}\boldsymbol{\Sigma}^{-\frac{1}{2}}$. Note that the loss is linear in $\boldsymbol{B}$ (and $\boldsymbol{B}'$). Since we restrict ourselves to $||\boldsymbol{A}||_2 \leq 1$, it is also true that $||\boldsymbol{B}||_2 \leq 1$.

We now show that the optimal $\boldsymbol{B}'$ has column space (and row space, since symmetric) lying in column space (and row space) of $\boldsymbol{M}'$. Let $\boldsymbol{B}^*$ be the optimal solution. Denote the eigendecomposition of $\boldsymbol{B}^*$ as $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, where $\boldsymbol{\Lambda} \succeq 0$ (as $\boldsymbol{B} \succeq 0$) and $\boldsymbol{U}$ is a unitary matrix. Equivalently :

$$\boldsymbol{B}^* = \sum_i \lambda_i\boldsymbol{u}_i\boldsymbol{u}_i^T \tag{5}$$

where $\boldsymbol{u}_i$ are columns of $\boldsymbol{U}$. Suppose the condition is not true. Then there exists a unit vector $\boldsymbol{v}$, such that $\boldsymbol{M'v} = 0$ and $\boldsymbol{B'v} \neq 0$.

We construct a new matrix $\bar{\boldsymbol{U}}$, whose columns are projection of $\boldsymbol{U}$ onto the plane with normal vector $\boldsymbol{v}$ given by $\boldsymbol{R} = \boldsymbol{I} - \boldsymbol{v}\boldsymbol{v}^T$. The projection matrix is defined such that $\boldsymbol{R}\boldsymbol{M'} = \boldsymbol{M'}$. Define $\bar{\boldsymbol{B}}$ from the constructed $\bar{\boldsymbol{U}}$.

$$\bar{\boldsymbol{U}} = \boldsymbol{R}\boldsymbol{U} = (\boldsymbol{I} - \boldsymbol{v}\boldsymbol{v}^T)\boldsymbol{U}$$
$$\bar{\boldsymbol{B}} = \bar{\boldsymbol{U}}\boldsymbol{\Lambda}\bar{\boldsymbol{U}}^T = \boldsymbol{R}\boldsymbol{B}^*\boldsymbol{R}^T$$

Now through some algebra we see that loss at $\bar{\boldsymbol{B}}$ is less than at $\boldsymbol{B}^*$ and hence $\boldsymbol{B}^*$ can't be optimal.

$$\begin{aligned}
&\langle \bar{\boldsymbol{B}}, (-\delta + \xi)\boldsymbol{M'} + \xi\boldsymbol{I} \rangle \\
=&\langle \boldsymbol{B}^*, (-\delta + \xi)\boldsymbol{R}^T\boldsymbol{M'}\boldsymbol{R} + \xi\boldsymbol{R}^T\boldsymbol{R} \rangle \\
=&\langle \boldsymbol{B}^*, (-\delta + \xi)\boldsymbol{M'} + \xi\boldsymbol{I} + \xi(\boldsymbol{R}^T\boldsymbol{R} - \boldsymbol{I}) \rangle \\
=&\langle \boldsymbol{B}^*, (-\delta + \xi)\boldsymbol{M'} + \xi\boldsymbol{I} \rangle + \xi\langle \boldsymbol{B}^*, (\boldsymbol{R}^T\boldsymbol{R} - \boldsymbol{I}) \rangle
\end{aligned}$$

Now we show that $\langle \boldsymbol{B}^*, (\boldsymbol{R}^T\boldsymbol{R} - \boldsymbol{I}) \rangle < 0$ and since $\xi > 0$, loss at $\bar{\boldsymbol{B}}$ is less than at $\boldsymbol{B}^*$. The fact is intuitively clear and aa formal proof is as follows :

$$\langle \boldsymbol{B}^*, \boldsymbol{R}^T\boldsymbol{R} \rangle = \mathrm{tr}(\boldsymbol{R}\boldsymbol{B}^*\boldsymbol{R}^T) = \sum_i \lambda_i ||\boldsymbol{R}\boldsymbol{u}_i||_2^2 < \sum_i \lambda_i ||\boldsymbol{u}_i||_2^2 = \langle \boldsymbol{B}^*, \boldsymbol{I} \rangle$$

The strict inequality is due to fact that $\boldsymbol{B'v} \neq 0$, i.e. there exists a i with $\lambda_i > 0$ and $\boldsymbol{v}^T\boldsymbol{u}_i \neq 0$ (and hence $||\boldsymbol{R}\boldsymbol{u}_i||_2^2 < ||\boldsymbol{u}_i||_2^2$). $||\boldsymbol{R}\boldsymbol{u}_j||_2^2 \leq ||\boldsymbol{u}_j||_2^2$ is true generally because $\boldsymbol{R}$ is a projection matrix.

Hence we showed that column space of $\boldsymbol{B}^*$ (i.e., optimal $\boldsymbol{B'}$) lies in column space of $\boldsymbol{M'}$. Now we show that it **spans the whole column space**. Suppose not. Let $\boldsymbol{B}^*$ be the optimal solution with decomposition with notation as used above. Then WLOG there exists $\boldsymbol{u}_i$ which $\in \mathrm{Span}(\boldsymbol{M'})$ and $\lambda_i = 0$ (if it doesn't exist we can rotate the $\boldsymbol{u}_j$'s with 0 eigenvalues so that a $\boldsymbol{u}_i$ aligns in the subspace). We take derivative w.r.t. λ_i and show that it's negative.

$$\begin{aligned}
\frac{\partial \mathcal{L}_{SS}}{\partial \lambda_i}\Big|_{\boldsymbol{B'}=\boldsymbol{B}^*} =& \frac{\partial \langle \boldsymbol{B'}, (-\delta + \xi)\boldsymbol{M'} + \xi\boldsymbol{I} \rangle}{\partial \lambda_i} \\
=& \boldsymbol{u}_i^T\big((-\delta + \xi)\boldsymbol{M'} + \xi\boldsymbol{I}\big)\boldsymbol{u}_i \\
=& (-\delta + \xi)\boldsymbol{u}_i^T\boldsymbol{M'}\boldsymbol{u}_i + \xi
\end{aligned}$$

if $\xi < \frac{\delta \boldsymbol{v}^T\boldsymbol{M'}\boldsymbol{v}}{1+\boldsymbol{v}^T\boldsymbol{M'}\boldsymbol{v}}$ for all directions $\boldsymbol{v}$ in $\mathrm{Span}(\boldsymbol{M'})$ (as $\boldsymbol{u}_i \in \mathrm{Span}(\boldsymbol{M'})$), then $\frac{\partial \mathcal{L}_{SS}}{\partial \lambda_i}$ is < 0. $\frac{\delta \boldsymbol{v}^T\boldsymbol{M'}\boldsymbol{v}}{1+\boldsymbol{v}^T\boldsymbol{M'}\boldsymbol{v}}$ is monotonic in $\boldsymbol{v}^T\boldsymbol{M'}\boldsymbol{v}$. The minimum value of $\boldsymbol{v}^T\boldsymbol{M'}\boldsymbol{v}$ is the smallest non-zero eigenvalue of $\boldsymbol{M'}$ denoted by $\lambda_{\min}$. Hence if $0 < \xi < \frac{\delta\lambda_{\min}}{1+\lambda_{\min}}$ we are good.

Now we showed that $\boldsymbol{B}^*$ (i.e., optimal $\boldsymbol{B'}$) spans the complete column space of $\boldsymbol{M'}$. Hence using the facts $\boldsymbol{B'} = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{B}\boldsymbol{\Sigma}^{\frac{1}{2}}$ and $\boldsymbol{B} = \boldsymbol{A}^T\boldsymbol{A}$, we can argue that for the optimal value of $\boldsymbol{A}$ denoted by $\boldsymbol{A}^*$, $\mathrm{Col}(\boldsymbol{A}^*) = S_F$. □

E Proof for CLIP Loss

In this section, we use the first part of Proposition B.1 to prove the theorem, i.e. we use the proof of $\mathrm{Col}(\boldsymbol{A}^*) \subseteq S_F$.

Proof. We write the objective by substituting the functional form of f as :

$$\begin{aligned}\mathcal{L} &= -\mathbb{E}_{\boldsymbol{x}_t,\boldsymbol{x}_v}[\boldsymbol{x}_t^T \boldsymbol{A}_t \boldsymbol{A}_v^T \boldsymbol{x}_v] \\ &\quad +\mathbb{E}_{\boldsymbol{x}_t}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}_v}[\exp(\boldsymbol{x}_t^T \boldsymbol{A}_t \boldsymbol{A}_v^T \tilde{\boldsymbol{x}}_v)])] \\ &= -\mathbb{E}_{\boldsymbol{x}_t,\boldsymbol{x}_v}[\boldsymbol{x}_t^T \boldsymbol{B} \boldsymbol{x}_v] + \mathbb{E}_{\boldsymbol{x}_t}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}_v}[\exp(\boldsymbol{x}_t^T \boldsymbol{B} \tilde{\boldsymbol{x}}_v)])]\end{aligned}$$

where $\boldsymbol{B} = \boldsymbol{A}_t \boldsymbol{A}_v^T$. We can further do a change of variable by $\boldsymbol{x}'_t = \boldsymbol{\Sigma}_t^{-\frac{1}{2}} \boldsymbol{x}_t$ and $\boldsymbol{x}'_v = \boldsymbol{\Sigma}_v^{-\frac{1}{2}} \boldsymbol{x}_v$. The means are now $\boldsymbol{\mu}'_{t,k} = \boldsymbol{\Sigma}_t^{-\frac{1}{2}} \boldsymbol{\mu}_{t,k}$ and $\boldsymbol{\mu}'_{v,k} = \boldsymbol{\Sigma}_v^{-\frac{1}{2}} \boldsymbol{\mu}_{v,k}$. Define $\boldsymbol{B}' = \boldsymbol{\Sigma}_t^{\frac{1}{2}} \boldsymbol{B} \boldsymbol{\Sigma}_v^{\frac{1}{2}}$. Now $\boldsymbol{x}'_t$ and $\boldsymbol{x}'_v$ have components with covariance being isotropic. Now we have :

$$\begin{aligned}\mathcal{L} &= -\mathbb{E}_{\boldsymbol{x}_t,\boldsymbol{x}_v}[\boldsymbol{x}_t^T \boldsymbol{B} \boldsymbol{x}_v] + \mathbb{E}_{\boldsymbol{x}_t}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}_v}[\exp(\boldsymbol{x}_t^T \boldsymbol{B} \tilde{\boldsymbol{x}}_v)])] \\ &= -\mathbb{E}_{\boldsymbol{x}'_t,\boldsymbol{x}'_v}[\boldsymbol{x}_t'^T \boldsymbol{B}' \boldsymbol{x}'_v] + \mathbb{E}_{\boldsymbol{x}'_t}[\log(\mathbb{E}_{\tilde{\boldsymbol{x}}'_v}[\exp(\boldsymbol{x}_t'^T \boldsymbol{B}' \tilde{\boldsymbol{x}}'_v)])]\end{aligned}$$

We argue that optimal $\boldsymbol{B}'^*$ has its row space equal to $\mathrm{Span}\{\boldsymbol{\mu}'_{v,k}\}$ and its column space equal to $\mathrm{Span}\{\boldsymbol{\mu}'_{t,k}\}$. We present the argument for column space (row space argument follows similarly).

Suppose there exists a unit vector $\boldsymbol{v}$ such that $\boldsymbol{v}^T \boldsymbol{\mu}'_{t,k} = 0$ for all $k \in [K]$, $\boldsymbol{v}^T \boldsymbol{B}'^* \neq 0$. Then we can define a new matrix as $\bar{\boldsymbol{B}}' = \boldsymbol{R}\boldsymbol{B}'^* = (I - 2\boldsymbol{v}\boldsymbol{v}^T)\boldsymbol{B}'^*$, where $\boldsymbol{R} = I - 2\boldsymbol{v}\boldsymbol{v}^T$ is a reflection matrix. Following arguments from Proposition B.1 we can prove that $\boldsymbol{x}_t'^T \boldsymbol{B}'^* \tilde{\boldsymbol{x}}'_v$ is identically distributed to $\boldsymbol{x}_t'^T \bar{\boldsymbol{B}}' \tilde{\boldsymbol{x}}'_v$. Hence using this we can show that $\mathcal{L}(\boldsymbol{B}'^*) = \mathcal{L}(\bar{\boldsymbol{B}}')$. Then by the strict convexity of the loss function $\mathcal{L}$ we have that $\mathcal{L}(\frac{\boldsymbol{B}'^* + \bar{\boldsymbol{B}}'}{2}) < \frac{\mathcal{L}(\boldsymbol{B}'^*) + \mathcal{L}(\bar{\boldsymbol{B}}')}{2} = \mathcal{L}(\boldsymbol{B}'^*)$. Hence $\boldsymbol{B}'^*$ can't be optimal.

Hence proved. □

F Representation collapse in non-contrastive learning

Consider a two component GMM in a d dimensional ambient space. The means of the components lie on the x and y axis (i.e. the first two dimensions), equidistant from the origin. Both components have isotropic covariance. We plot the first two dimensions below in Figure 3. Note that for this setup

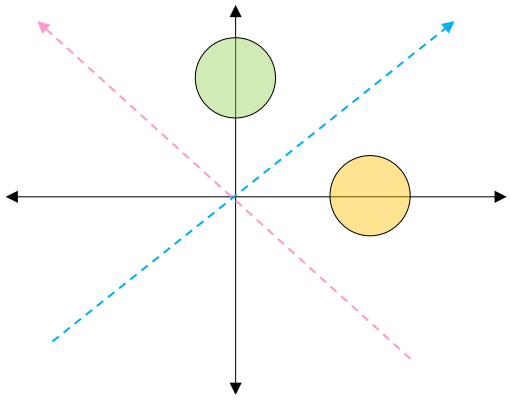
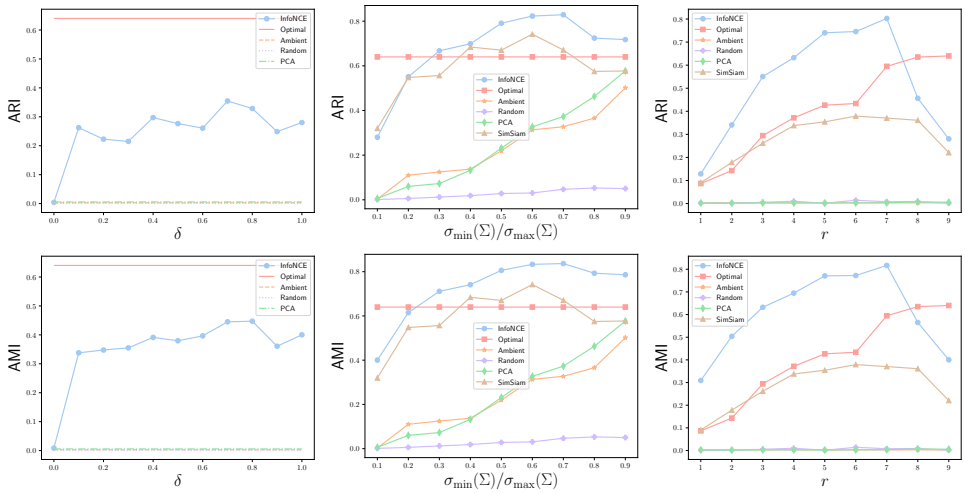


Figure 3: The means of the GMM components in the first two dimensions.

the rank of the fisher subspace is two (i.e., $K = 2$) and the fisher subspace is the $x - y$ plane.

Consider learning a mapping matrix onto a one-dimensional subspace (i.e. $r < K$). InfoNCE type contrastive objectives would learn the subspace given by the pink line (i.e. gaussian would be projected onto the pink line and hence would be well seperated). This is stated without proof, but we know that InfoNCE loss would learn a subspace lying in the fisher subspace. And through some basic algebra we can convince ourselves that the solution would be the pink like (i.e. the line $x + y = 0$).



(a) Noise in augmentation (b) Variance orthogonal to S_F (c) Dimension of projected space

Figure 4: We further extend the results presented in Figure 2 with orthonormalized InfoNCE and SimSiam mappings

But for non-contrastive objectives like SimSiam, while we can only prove that the optimal solution lies in the $x - y$ plane. SimSiam objective is not able to distinguish between the lines $x + y = 0$ and $x = y$, and hence might lead to collapse of representations. But as stated in the Theorem 4.1, this is only the case if $r < K$. For $r \geq K$, the SimSiam objective learns the complete Fisher subspace (and only the fisher subspace).

G Experimental Details and Additional Results

G.1 Data Generation for Synthetic Experiments

Ambient Dimension We consider the ambient dimension to be 100 for all experiments (except for scaling curves in Figure 2d, where we let ambient dimension equal to $K - 1$).

Means of Components For generating means of the components, we first sample $K - 1$ times from a 0 mean, $\boldsymbol{I}$ normal distribution in ambient dimension space. We chose the Kth mean such that the means sum upto 0. Hence the means lie in a $K - 1 = 9$ dimensional space.

Covariance Matrix To construct a covariance matrix we start with a unit variance $\boldsymbol{I}$ matrix. We then upscale the variance in (Ambient - NumMeans+1) dimensional space orthogonal to the means subspace by factor of κ, where $\kappa = \sigma_{\max}(\boldsymbol{\Sigma})/\sigma_{\min}(\boldsymbol{\Sigma})$. We take κ to be $1/0.1 = 10.0$ by default. For scaling experiments, we choose random NumMeans/2 orthogonal directions, and increase the covariance in those directions. Note for condition number experiment we take κ to be $1/0.x$ where x goes from 1 to 9.

G.2 Orthonormalized InfoNCE and SimSiam

We plot the ARI/AMI numbers for orthonormalized InfoNCE and SimSiam matrices in Figure 4. Specifically, we do a QR decomposition on these matrices and take the orthogonal Q matrix as the projection matrix. In these plots the convergence issues of InfoNCE loss become more apparent, as we can see that InfoNCE lagging behind optimal.

G.3 CIFAR-100 Experiments and Additional Results

In this section, we present a seeded version of our results in the manuscript for the CIFAR-100 experiments. We randomly sample 20 classes from the datasets and collect the data points that belong to these 20 classes. The randomly selected subset of images are then normalized before inputting the the mapping method. Once the methods learn linear maps, it is then applied on the normalized data.

Table 3: **Clustering performance of linear dimensionality reduction methods on CIFAR100: Bolded** and underlined values indicate the best and second-best scores for each column. We report clustering performance for K-Means on $r = 5, 10, 15, 19$-dimensional subspaces (mapping dimensions) using the linear mappings learned by 6 different methods. We measure the clustering performance using ARI and AMI. We run the methods for 5 different seeds, corresponding to 5 random 20-class subsets of CIFAR-100 datasets. The average scores over 5 runs are reported with standard deviation given in paranthesis. InfoNCE and LDA outperform the rest by significant margins in all mapping dimensions. LDA shows the best performance in terms of ARI, while InfoNCE achieves higher AMI values across the board.

	Mapping Dim = 5		**Mapping Dim = 10**		**Mapping Dim = 15**		**Mapping Dim = 19**	
Method	**ARI**	**AMI**	**ARI**	**AMI**	**ARI**	**AMI**	**ARI**	**AMI**
Random	0.020 (0.003)	0.061 (0.009)	0.026 (0.003)	0.078 (0.009)	0.028 (0.003)	0.084 (0.007)	0.033 (0.007)	0.094 (0.015)
Ambient	0.039 (0.006)	0.113 (0.013)	0.039 (0.006)	0.113 (0.013)	0.039 (0.006)	0.113 (0.013)	0.039 (0.006)	0.113 (0.013)
PCA	0.038 (0.006)	0.110 (0.013)	0.040 (0.006)	0.113 (0.014)	0.040 (0.006)	0.114 (0.014)	0.041 (0.007)	0.116 (0.015)
SimSiam	0.035 (0.009)	0.111 (0.020)	0.037 (0.009)	0.113 (0.021)	0.035 (0.010)	0.110 (0.021)	0.035 (0.009)	0.110 (0.018)
InfoNCE	0.058 (0.009)	**0.146** (0.016)	0.064 (0.011)	**0.156** (0.018)	0.063 (0.005)	**0.157** (0.014)	0.066 (0.008)	**0.159** (0.014)
LDA	**0.060** (0.008)	0.145 (0.014)	**0.074** (0.012)	0.153 (0.013)	**0.074** (0.010)	0.148 (0.014)	**0.076** (0.014)	0.149 (0.017)

Finally, we cluster the (linearly) transformed data using K-Means where the number of classes is the same as the number of components in the GMM. We repeat this procedure for 5 times with different seeds for the random number generator. We report the results in Table 3, which are consistent with the results in the main text.

H Linear Discriminant Analysis (LDA) and Fisher LDA

H.1 Linear Discriminant Analysis

The classical binary class LDA objective is defined as a bayes-optimal solution for classification under the assumption that the data is generated from a two-component Gaussian mixture model with identical covariances. For non-shared covariance for two component case we don't have a closed form solution and the problem is referred to as Quadratic discriminant Analysis. The LDA solution for two-component GMM with shared covariance is the subspace projection on which leads to no loss in likelihood of data. It is given as :

$$\mathcal{S}_{LDA} = \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

H.2 Fischer Linear Discriminant Analysis

Instead of defining the LDA objective to be the Bayes-optimal under the assumption that data is generated from a two-component Gaussian mixture model with shared covariance, Fischer LDA considers an alternative objective. It does away with both the GMM and the shared covariance assumption and assumes data to be coming from two different distributions, where each distribution belonged is defined by it's class. Fisher defined the separation between these two distributions to be the ratio of the variance between the classes to the variance within the classes i.e.

$$\mathcal{S}_{fischer} = \frac{|\boldsymbol{\theta}^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)|^2}{|\boldsymbol{\theta}^T(w_1\mathbf{\Sigma}_1 + w_2\mathbf{\Sigma}_2)\boldsymbol{\theta}|}$$

The solution maximizing this is given by $(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ is termed the Fisher subspace. When the data is generated from a two-component shared covariance GMM, the solution coincides with $\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ learnt by LDA.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We propose a standard theory framework in machine learning, and provide theoretical results that validate our claims. We also conduct experiments to corroborate out theoretical claims.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have a section for limitations at the end of the paper. We also follow up the theorem statements with discussions on the implications and shortcomings of the results.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We clearly state the assumptions we make for the theorems. In particular, we refer to the data and augmentation distribution assumptions and clearly define the objective functions with their properties.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We state the details of our experimental setup. For the synthetic experiments, we give the recipe for data generation and for the real data, we describe how the data is randomly subsampled.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We are planning to release our code after we prepare a clean and distributable version which is also de-anonymized.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe the experimental setup in details in the manuscript. We provide details on data generation, split and also properties of the algorithm(s) used.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Please see the Appendix for a complete overview of our results with statistical details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss the details related to compute resources in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We abide by the NeurIPS Code of Ethics to the best of our abilities.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: We study a fundamental theory problem in machine learning. Our work and results do not have any negative societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not pose any such risks related to its content; we do not release new models or datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Our theory and numerical experiments do not use any existing assets that requires license. Any academic resource, model, dataset and algorithm used are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release any assets that requires licensing and/or documentation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The theory-oriented scope of our work does not require such approvals.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The development process does not involve LLM use as any important part of it.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.