An ideal learned representation should display transferability and robustness. Supervised contrastive learning (SupCon) is a promising method for training accurate models, but produces representations that do not capture these properties due to class collapse—when all points in a class map to the same representation. Recent work suggests that “spreading out” these representations improves them, but the precise mechanism is poorly understood. We argue that creating spread alone is insufficient for better representations, since spread is invariant to permutations within classes. Instead, both the correct degree of spread and a mechanism for breaking this invariance are necessary. We first prove that adding a weighted class-conditional InfoNCE loss to SupCon controls the degree of spread. Next, we study three mechanisms to break permutation invariance: using a constrained encoder, adding a class-conditional autoencoder, and using data augmentation. We show that the latter two encourage clustering of latent subclasses under more realistic conditions than the former. Using these insights, we show that adding a properly-weighted class-conditional InfoNCE loss and a class-conditional autoencoder to SupCon achieves 11.1 points of lift on coarse-to-fine transfer across 5 standard datasets and 4.7 points on worst-group robustness on 3 datasets, setting state-of-the-art on CelebA by 11.5 points.

1. Introduction

Learning a representation with a favorable geometry is a critical challenge for modern machine learning. Good geometries can engender strong downstream transfer performance and robustness to subgroup imbalances, whereas poor geometries may have low transferability and be brittle (Islam et al., 2021; Sohoni et al., 2020). However, producing—or even characterizing—a good geometry can be difficult.

We focus on the challenges in doing so with supervised contrastive learning (SupCon). SupCon is a promising method for training accurate machine learning models (Khosla et al., 2020), but suffers from class collapse—wherein each point in the same class has the same representation, as in Figure 1 far left (Graf et al., 2021). Collapsed representations cannot distinguish fine-grained details within classes—in particular latent subclasses—resulting in poor transferability and robustness. Modifications to SupCon that heuristically “spread out” its representations have shown empirical promise (Islam et al., 2021), but a precise understanding of spread—how separated individual points are in representation space—and how to control it is lacking.

Furthermore, spread alone is not sufficient to explain improved representations. We observe that modifications to SupCon that increase spread are invariant to class-fixing permutations. That is, the loss value does not change when points of the same class are arbitrarily permuted in representation space. For example, Figure 1 right visualizes two geometries that both have spread but differ in representation quality, as suggested by the significant gap in transfer learning performance (35.4 points). Thus, while spread may be important, another mechanism is needed to break class-fixing permutation invariance for good performance.

We argue that these are the two key challenges to improving SupCon’s representations: creating the correct degree of spread, and breaking class-fixing permutation invariance. This paper makes progress on these challenges.

Challenge 1: Balancing Spread. We first prove a simple result that a class-collapsed representation cannot have good transfer performance, which motivates spread. We then analyze whether \( L_{\text{spread}} \), a loss function that combines SupCon with a class-conditional InfoNCE loss, can induce spread.

We find that previous approaches for analyzing contrastive losses encounter a technical challenge because SupCon
and InfoNCE have incompatible optimal geometries (class collapse and uniformity on the hypersphere, respectively). For example, Wang & Isola (2020) analyze individual loss components in isolation, but doing so risks drawing misleading conclusions when the loss components are incompatible. Further, finding exact solutions to optimization problems on the hypersphere is fundamentally difficult; a classic example is the Thomson problem (Thomson, 1897), which has evaded an exact solution after a century of study.

We bypass these problems by constructing a distribution that is neither collapsed nor uniform and analyzing its loss. We introduce \( s_f(y) \), a notion of class variance, to measure spread on this distribution. We show that this distribution has an intermediate degree of spread by deriving bounds for the weight \( \alpha \) on the class-conditional InfoNCE loss within which this distribution attains lower loss than either extreme. While this result does not fully characterize the geometry, it suggests that setting \( \alpha \) properly can induce an optimal distribution with appropriate spread—which we validate with measurements on CIFAR10.

**Challenge 2: Breaking Permutation Invariance.** Our first result demonstrates that \( L_{\text{spread}} \) can induce spread but does not give insight on class-fixing permutation invariance. We formally define class-fixing permutation invariance and prove that \( L_{\text{spread}} \) is subject to it absent other interventions.

This motivates the question: how should we break class-fixing permutation invariance? We show that inducing an inductive bias towards clustering of latent subclasses can break permutation invariance—and more importantly, can result in good coarse-to-fine transfer performance. We introduce \( \sigma_f(z) \), a measure of subclass clustering in representation space, and show that coarse-to-fine generalization error scales with \( \sigma_f(z)/s_f(y) \).

A standard approach to controlling \( \sigma_f(z) \) is assuming Lipschitzness of the model. However, Lipschitzness is a strong assumption for modern deep networks, which are powerful enough to memorize random noise (Zhang et al., 2016). In empirical measurements, we find that modern deep networks display poor Lipschitzness, and thus the Lipschitzness assumption is insufficient for inducing clustered subclass representations.

We thus propose two alternatives that can bound \( \sigma_f(z) \) under more realistic assumptions: directly encoding fine-grained details by concatenating the representations from a class-conditional autoencoder, and using data augmentation in the class-conditional InfoNCE loss. The former only requires a “reverse Lipschitz” decoder to upper bound \( \sigma_f(z) \), and can do so by a constant factor tighter than a general (non-conditional) autoencoder. The latter only requires the encoder to be Lipschitz over data augmentations to induce subclass clustering—and can also explain observations from prior work (Islam et al., 2021). We validate these findings by measuring Lipschitzness constants and \( \sigma_f(z)/s_f(y) \) on real data; we find that these alternate assumptions are more realistic than overall Lipschitzness, and that data augmentation and autoencoders help induce subclass clustering.

**Empirical Validation** Using our theoretical insights, we propose THANOS: adding a class-conditional InfoNCE loss and a class-conditional autoencoder to SupCon. We evaluate THANOS on two tasks designed to evaluate how well it preserves subclasses:

- **Coarse-to-fine transfer learning** trains a model to classify superclasses but use the representations to distinguish subclasses. THANOS outperforms SupCon by...
We have a class-balanced labeled training dataset. We assume that the data is class-balanced such that
\[ D \]
variable
and the value of each
\[ y \]
Coarse-to-fine transfer eval-
Contrastive learning trains an encoder to produce representa-

2. Background
Section 2.1 presents our data model and the coarse-to-fine transfer task. Section 2.2 presents \( L_{\text{spread}} \), a simple variant of SupCon that adds a weighted class-conditional InfoNCE loss. Section 2.3 discusses geometry of contrastive losses.

2.1. Data Setup
Input data \( x \in X \) are drawn from a distribution \( P \) with deterministic class \( y = h(x) \), where \( y \in Y \equiv \{0, \ldots, K-1\} \). We assume that the data is class-balanced such that \( \Pr(y=i) = \frac{1}{K} \) for all \( i \in Y \).

Data points also belong to latent subclasses. Following Sohoni et al. (2020), we denote a subclass as a latent discrete variable \( z \in Z \). \( Z \) can be partitioned into disjoint subsets \( S_0, \ldots, S_{K-1} \) such that if \( z \in S_k \), then its corresponding y label is equal to \( k \). For simplicity, we assume that there are two subclasses for each label \( k \), e.g. \( |S_k| = 2 \). The data generating process proceeds as follows: first, the latent subclass \( z \) is sampled with proportion \( p(z) \). Then, \( x \) is sampled from the distribution \( P_x = p(z) \), and its corresponding deterministic label is denoted \( y = S(z) \). Let \( h_s(x) : X \to Z \) denote \( x \)’s subclass.

We have a class-balanced labeled training dataset \( D = \{ (x_i, y_i) \}_{i=1}^{m} \) where points \( (x_i, z_i, y_i) \) are drawn i.i.d, and the value of each \( z_i \) is unknown during training time. Denote \( D_y = \{ x \in D : h(x) = y \} \) and \( D_z = \{ x \in D : h_s(x) = z \} \), and denote their sizes by \( n_y = |D_y| \) and \( n_z = |D_z| \).

Contrastive learning trains an encoder \( f : X \to R^d \) on \( D \) that maps inputs to representations in an embedding space \( R^d \).

Coarse-to-Fine Transfer
Coarse-to-fine transfer evaluates how well an embedding trained on coarse classes \( Y \) distinguishes fine classes (subclasses) \( Z \). Fix a \( y \) and make-uppose that \( S_y = \{ z, z' \} \). The task is to classify \( z \) versus \( z' \) using the encoder \( f \) learned on \( D \). We are given a dataset of subclass labels, \( D_z = \{ (x_i, z) \}_{i=1}^{m} \). Denote \( D_{x,z} = \{ x \in D_x : h_s(x) = z \} \) and \( n_z = |D_{z,s}| \). We learn linear weights \( W_z, W_{z'} \in R^d \) and construct an estimate \( \hat{p}(z|f(x)) \) by using softmax scores \( \hat{p}(z|f(x)) = \frac{\exp(f(x) \cdot W_z)}{\exp(f(x) \cdot W_z) + \exp(f(x) \cdot W_{z'})} \), where \( f \) is fixed. We use the mean classifier to construct \( W \), following prior work (Arora et al., 2019). That is, \( W_z = \frac{1}{m} \sum_{x \in D_{x,z}} f(x) \), and \( W_{z'} \) is similarly defined.

We evaluate the performance of coarse-to-fine transfer with a \( \gamma \)-margin loss, defined on a point \( (x, z) \) as
\[ \ell_{\gamma}(f(x, z)) = 1 - \mathbb{I} \{ \hat{p}(z|f(x)) \geq \gamma \hat{p}(z'|f(x)) \} \]
for \( \gamma > 1 \). That is, we want the model to output the correct subclass label at least \( \gamma \) times more likely than the incorrect one. Define the \( \gamma \)-margin generalization error on \( z \) as
\[ L_{\gamma}(f(z)) = \mathbb{E}_{x \sim \mathcal{P}_z}[\ell_{\gamma}(f(x, z))] \].

2.2. A Modified Supervised Contrastive Loss
Contrastive learning trains an encoder to produce representations of the data by pushing together similar points (positive pairs) and pulling apart different points (negative pairs). We consider \( L_{\text{spread}} \), a weighted sum of a supervised contrastive loss \( L_{\text{sup}} \) (Khosla et al., 2020) and a class-conditional InfoNCE loss \( L_{\text{Cnce}} \).

Let \( B \) be a batch of data from \( D \). Define \( P(i,B) = \{ x^+ \in B \mid h(x^+) = h(x_i) \} \) as the points with the same label as \( x_i \) and \( N(i,B) = \{ x^- \in B \mid h(x^-) \neq h(x_i) \} \) as points with a different label. Let \( a(x_i) \) be an augmentation of \( x_i \), and assume that augmentations of each sample are disjoint. Denote \( \sigma_f(x, x') = \exp(f(x)^\top f(x'))/\tau \) with temperature hyperparameter \( \tau \). For \( \alpha \in [0,1] \), \( \hat{L}_{\text{spread}}(f, x, B) \) on \( x \) belonging to \( B \) is:
\[ \hat{L}_{\text{spread}}(f, x, B) = (1 - \alpha) L_{\text{sup}}(f, x, B) + \alpha L_{\text{Cnce}}(f, x, B) \]
where
\[ L_{\text{sup}}(f, x, B) = -1 / |P(i,B)| \times \sum_{x^+ \in P(i,B)} \log \frac{\sigma_f(x_i, x^+)}{\sigma_f(x_i, x^-) + \sum_{x^- \in N(i,B)} \sigma_f(x_i, x^-)} \]
\[ L_{\text{Cnce}}(f, x, B) = -\log \sum_{x^+ \in P(i,B)} \sigma_f(x_i, x^+) \]
The overall loss \( \hat{L}_{\text{spread}}(f, B) \) is averaged over all points in \( B \). \( L_{\text{sup}} \) is a variant of the SupCon loss (Khosla et al., 2020). \( L_{\text{Cnce}} \) is a class-conditional version of the InfoNCE loss, where the positive distribution consists of augmentations and the negative distribution consists points from the same class, intuitively encouraging them to be spread apart.

2.3. Geometries of Contrastive Losses
We present a series of standard theoretical assumptions for analyzing contrastive geometry, and define two important distributions—class collapse and class uniformity.

Assumptions
We make several standard theoretical assumptions (Graf et al., 2021; Wang & Isola, 2020; Robinson et al., 2020): 1) restrict the encoder \( f \)’s output space to be \( S^{d-1} \), the unit hypersphere (i.e. normalized outputs); 2) assume that \( K \leq d + 1 \), such that a \( K \)-regular simplex inscribed in \( S^{d-1} \) exists; 3) assume that the encoder is infinitely powerful,
meaning that any distribution on $S^{d-1}$ is realizable by $f(x)$. We define the pushforward measure of the class-conditional distribution of $p(\cdot| h(x) = y)$ via $f$ as $\mu_y$ for $y \in \mathcal{Y}$, where $\mu_y \in \mathcal{M}(S^{d-1})$ is over all Borel probability measures on the hypersphere. Define $\mu = \{\mu_y\}_{y \in \mathcal{Y}}$ as the overall pushforward measure corresponding to $\mathcal{P} \circ f^{-1} \in \mathcal{M}(S^{d-1})$.

**Class Collapse Distribution** Define $\nu = \{v_y\}_{y \in \mathcal{Y}} \in S^{d-1}$ as the set of vectors forming the regular simplex inscribed in the hypersphere, satisfying: a) $\sum v_y = 0$; b) $\|v_y\|_2 = 1 \forall y$; and c) $\exists c_K \in \mathbb{R}$ s.t. $v_y^Tv_y' = c_K$ for $y \neq y'$. Let $\delta_v$ be the probability measure on $S^{d-1}$ with all mass on $v_y$, and let $\delta_{\nu}$ be the class-collapsed measure such that $\mu_y = \delta_{\nu}$ and $f(x) = v_y$ almost surely whenever $h(x) = y$. Graf et al. (2021) show that $\mu = \delta_{\nu}$ minimizes the SupCon loss.

**Class Uniform Distribution** Denote $\sigma_{d-1}$ as the normalized surface area measure on $S^{d-1}$. $\mu = \sigma_{d-1}$ is the class-uniform measure when $\mu_y = \sigma_{d-1}$ for all $y \in \mathcal{Y}$. Wang & Isola (2020) show that $\sigma_{d-1}$ minimizes the InfoNCE loss.

### 3. Controlling Spread

In Section 3.1, we demonstrate the importance of spread—having distinguishable representations of points in a class—by showing that SupCon results in poor coarse-to-fine transfer. In Section 3.2, we begin to explore whether $L_{spread}$ can result in more spread out geometries. We define the asymptotic form of $L_{spread}$ and apply the approach from Wang & Isola (2020) to analyze individual loss terms. We find that the optimal geometries of the individual terms are incompatible. In Section 3.3, we analyze the asymptotic $L_{spread}$ as a whole using a nuanced approach that compares the loss over different geometries. We conclude that the optimal geometry is neither class-collapsed nor class-uniform for a range of $\alpha$. This result suggests that spread can be carefully controlled, and we capture this property by introducing a notion of intra-class variance, $s_f(y)$. All proofs for the paper are in Appendix C.

#### 3.1. The Importance of Spread

SupCon exhibits class collapse on the training data and does not spread out representations in a class. We use standard generalization bounds and show that this geometry results in poor coarse-to-fine generalization error: asymptotically, the error obtains its maximum possible value.

Define $f_{SC} \in \mathcal{F}$ to be the encoder trained with SupCon satisfying class collapse, $f_{SC}(x) = v_y$ for all $x \in \mathcal{D}$ where $h(x) = y$. Let $f(x)[j]$ be the $j$th entry of $f(x)$. For function class $\mathcal{F}$, let $F_j = \{f(\cdot)[j] : f \in \mathcal{F}\}$ be the elementwise class. Let $\mathcal{R}_n(F_j)$ denote $F_j$’s Rademacher complexity on $n$ samples, and define $\mathcal{R}_n(\mathcal{F}) = \sum_{j=1}^{n} \mathcal{R}_n(F_j)$.

**Theorem 1.** For $\gamma$ where $\log \gamma \geq 8\max \{\mathcal{R}_{n_z}(\mathcal{F}), \mathcal{R}_{n_y}(\mathcal{F})\}$, SupCon’s coarse-to-fine error is at least $L_{L_{\gamma}f_{SC}}(z) \geq 1 - \delta(n_z, \mathcal{F}, \gamma) - \delta(n_z, \mathcal{F}, \gamma) - \xi(m_z \wedge m_z', \gamma)$, where $\delta(n_z, \mathcal{F}, \gamma) = \exp\left(-\frac{m_z}{2\gamma^2}(\log \gamma - 8\mathcal{R}_{n_z}(\mathcal{F})^2)\right)$ bounds generalization error of $f_{SC}$ and $\xi(m_z \wedge m_z', \gamma) = \exp\left(-\frac{(m_z \wedge m_z')\log \gamma}{n_z}\right)$ bounds the noise from $D_x$.

As $n \wedge m$ increases, error approaches 1—its maximum value—and the model will almost surely predict the correct subclass $\gamma$ times less often than the incorrect one. This result motivates studying whether $L_{spread}$ can encourage spread.

#### 3.2. Asymptotic $L_{spread}$

We present the asymptotic version of $L_{spread}$. For a given anchor $x \sim \mathcal{P}$, define a positive pair $x^+ \sim p(\cdot| h(x^+) = h(x))$ from the same class and a negative pair using $x^- \sim p(\cdot| h(x^-) \neq h(x))$ from a different class. Let $a(x)$ be an augmentation of $x$ drawn from a distribution $p_a(\cdot|x)$, where each $p_a(\cdot|x)$ has disjoint support.

**Definition 1.** Define $L_{spread}(f, \alpha)$ as

$$L_{spread}(f, \alpha) = (1-\alpha)\ L_{align}(f) + \alpha\ L_{aug}(f) + \alpha\ L_{same}(f)$$

where

$$L_{align}(f) = \mathbb{E}_{x,x^+} \left[ \left\| f(x) - f(x^+) \right\|^2 / 2\tau \right]$$

$$L_{aug}(f) = \mathbb{E}_{x,a(x)} \left[ \left\| f(x) - f(a(x)) \right\|^2 / 2\tau \right]$$

$$L_{diff}(f) = \mathbb{E}_x \left[ \log \mathbb{E}_{x^-} \left[ \exp\left(-\left\| f(x) - f(x^-) \right\|^2 / 2\tau \right) \right] \right]$$

$$L_{same}(f) = \mathbb{E}_x \left[ \log \mathbb{E}_{a(x)} \left[ \exp\left(-\left\| f(x) - f(a(x)) \right\|^2 / 2\tau \right) \right] \right]$$

The derivation of $L_{spread}(f, \alpha)$ is in Appendix C.1. Next, we analyze individual terms, similar to Wang & Isola (2020)’s approach. For simplicity, we present the binary setting $K = 2$. We abuse notation and use $f$ and $\mu$, the pushforward measure of $x$ on the hypersphere, interchangeably in $L_{spread}(f, \alpha)$ as well as in the loss components in Definition 1.

**Proposition 1 (Individual losses).** $L_{align}(f)$ and $L_{aug}(f)$ are minimized when $f(x) = f(x^+)$ and $f(x) = f(a(x))$ almost surely, respectively. $L_{diff}(\mu)$ is minimized when $\mu = \delta_{\nu}$. $L_{same}(\mu)$ is minimized when $\mu = \sigma_{d-1}$.

When $\alpha = 0$, the “active” loss terms are $L_{align}$ and $L_{diff}$, whose optima are jointly realizable and yield $\mu = \delta_{\nu}$ overall. When $\alpha = 1$, the terms $L_{aug}$ and $L_{same}$ are also compatible, yielding $\mu = \sigma_{d-1}$ and augmentations with the same embedding as their original point.

Neither of these distributions has good coarse-to-fine transfer performance on its own: $\delta_{\nu}$ loses information within classes, and $\sigma_{d-1}$ allows points of different classes to be close together (Figure 1 left). To avoid both $\delta_{\nu}$ and $\sigma_{d-1}$, $\alpha \in (0, 1)$ must achieve a balance between the two loss terms. But the behavior of the weighted loss overall is unclear from the result in Proposition 1. It is also unclear whether there even exists an intermediate distribution that minimizes $L_{spread}(\mu, \alpha)$. 

3.3. Our Result

We seek to analyze the geometry of the overall loss. Explicitly characterizing the optimal geometry is challenging, so we design a family of measures on the hypersphere and examine when such measures obtain lower loss than collapsed or uniform measures. We perform analysis for $K = 2$ and consider $K = 3$ in Appendix D. Synthetic experiments are in Appendix H.

The measure we study, $\mu_\theta$, assigns mass evenly on two points that are close to $y_\theta$, a vertex of the regular simplex, but separated by some angle $\theta$ for each $\mu_y$ (see Figure 4 in Appendix H). Formally, define a block-diagonal rotation matrix $R_\theta \in \mathbb{R}^{d \times d}$ consisting of submatrices $[^{\cos \theta}_{\sin \theta} \ -^{\sin \theta}_{\cos \theta}]$ and $I_{d-2}$ on the diagonal. For $\theta \in (0, \pi/2)$, define the measure $\mu_\theta = \{\mu_0, \mu_1, \mu_\theta\}$, where $\mu_0, \mu_1 = \frac{1}{2} \delta_{R_{\pi/2}, 0} + \frac{1}{2} \delta_{R_{\theta}, 0}$, and similarly $\mu_\theta = \frac{1}{2} \delta_{R_{\theta}, 0} + \frac{1}{2} \delta_{R_{-\theta}, 0}$. We present a technical result on the range of $\alpha$ for which $\mu_\theta$ attains lower loss than both class-collapsed or class-uniform measures.

**Theorem 2.** Let $c_{r,d} = 2^{d+1} - \sqrt{2}\left(-\frac{2d+1}{2} - 2\log W_{1/2}(S^{d-1})\right)$, where $W_{1/2}(S^{d-1})$ is a constant depending on $d$ and $(d, 2)$ (see Appendix C.1 for exact value). Then, when $\alpha \in (2/3, c_{r,d})$

$$\theta^* = \arcsin \sqrt{2 \log \frac{3w-3}{3w}}$$

minimizes $L_{\text{spread}}(\mu_\theta, \alpha)$ and satisfies $L_{\text{spread}}(\mu_\theta, \alpha) \leq \min_{\mu \in \{\delta_\sigma, \sigma_{d-1}\}} L_{\text{spread}}(\mu, \alpha)$.

Our result does not define the exact optimal geometry since it constrains the measures we optimize to be over $\mu_\theta$. For $\alpha \notin (2/3, c_{r,d})$, it also does not specify the optimal geometry—we only know that the optimal geometry is not of form $\mu_\theta$.

However, our result yields a high-level insight: there exists a range of $\alpha$ for which the optimal geometry that minimizes $L_{\text{spread}}(\mu, \alpha)$ spreads out points on the hypersphere. Concretely, define the spread of class $y$ under $f$ as $s_f(y) = \mathbb{E}_{h(x)=y} ||f(x) - \mathbb{E}_{h(x)=y}[f(x)]||$.

**Corollary 1.** If $\alpha \in (2/3, c_{r,d})$ and $f(x)$ has measure $\mu_\theta^*$, the spread for $y$ under $f$ is $s_f(y) = \sqrt{\frac{2}{3} \log \frac{3w-3}{3w}} \sim \omega(1)$.

In other words, $L_{\text{spread}}$ can yield an extent of spread $s_f(y)$ that is controlled by $\alpha$. Experiments on CIFAR10 support our result (Figure 2); the geometry is collapsed for low values of $\alpha$, followed by a region of spread, followed by uniformity.

Finally, we remark on two deliberate aspects of our analysis. First, to avoid issues of non-convexity, we directly compare the overall loss of our measures with those of the two extrema $\delta_\sigma$ and $\sigma_{d-1}$. Second, general distributions beyond the regular simplex and normalized surface measure are hard to compute contrastive losses over, and such computations are not largely studied to the extent of our knowledge. This inherently restricts analysis to simple distributions like $\mu_\theta$.

### 4. Breaking Permutation Invariance

Our analysis in the previous section shows that $L_{\text{spread}}$ can obtain an optimal geometry that is neither collapsed nor uniform. However, this result does not completely explain improved transfer performance because $L_{\text{spread}}$ under the previous setup is class-fixing permutation invariant, a property we define in Section 4.1. Inducing an inductive bias can break such an invariance. We argue that an inductive bias that encourages clustering of latent subclasses can be particularly useful. In Section 4.2, we show that generalization error on coarse-to-fine transfer learning depends on both $s_f(y)$ and a notion of subclass clustering $\sigma_f(z)$. We thus discuss three approaches for controlling $\sigma_f(z)$: one standard, and two alternatives with more realistic assumptions (Section 4.3).

#### 4.1. Class-Fixing Permutation Invariance

First, we define class-fixing permutation invariance.

**Definition 2** (Class-Fixing Permutation Invariance). Let $F$ be a class of encoders. Let $L(f, B)$ be a loss function over an encoder $f \in F$ and a set of $n$ points $B = \{x_1, \ldots, x_n\}$. Define $S_{h,B}$ as the set of class-fixing permutations such that $\pi \in S_{h,B}: [n] \rightarrow [n]$ satisfies $h(x_{\pi(i)}) = h(x_i)$ for all $i \in [n]$. Then, $L$ is invariant on class-fixing permutations under $F$ if, for any batch $B$, permutation $\pi \in S_{h,B}$, and encoder $f \in F$, there exists another encoder $f^* \in F$ such that $f^*(x_i) = f(x_{\pi(i)})$ for all $i \in [n]$ and $L(f, B) = L(f^*, B)$.

We find that $L_{\text{spread}}$ is invariant on class-fixing permutations under the infinite encoder assumption from Section 2.3.

**Proposition 2.** Let $F$ be the set of infinite encoders. Then $L_{\text{spread}}$ is invariant on class-fixing permutations under $F$.

Under class-fixing permutation invariance, data points can be arbitrarily mapped to representations within their classes, suggesting that the mapping that minimizes $L_{\text{spread}}$ is not unique. However, not all these mappings achieve the same performance on downstream tasks. Therefore, while our...
result from Section 3 provides insight into $L_{\text{spread}}$’s geometry under an infinitely powerful encoder, it cannot completely explain representation quality.

### 4.2. Inductive Bias for Improved Coarse-to-fine Transfer

Inducing an inductive bias can break permutation invariance (see Lemma 4 in Appendix D for a simple proof of how smoothness of $f$ is a sufficient condition for breaking invariance). We argue that inducing subclass clustering can be particularly helpful for transfer performance. We measure subclass clustering in embedding space via the expected distance to the center of the subclass, $\sigma_f(z) = E_{x \sim p_x} || f(x) - E_{x \sim p_x} [f(x)] ||$. We show that this quantity $\sigma_f(z)$, along with degree of spread $s_f(y)$, is critical for the generalization error of coarse-to-fine transfer.

To present our result on coarse-to-fine generalization error, we define some additional terms. Let $y$ denote the class label corresponding to $z, z'$. Define the quantity $\delta_f(z, z') = \frac{1}{p(z) |p(z) - p(z')|^2} | \sigma_f(z) - p(z')|^2 \sigma_f(z')$ as a notion of separation between $z$ and $z'$. $\delta_f(z, z')$ is large when there is spread (large $s_f(y)$) and sufficient subclass clustering (low $\sigma_f(z), \sigma_f(z')$). Define the variance of a subclass as $\text{Var}_f[z] = E_{x \sim p_x} [|| f(x) - E_{x \sim p_x} [f(x)] ||^2]$. We assume that for all $z \sim p_z$, there exists a $c > 0$ such that $| f(x) - E_{x \sim p_x} [f(x)] | \geq c E_{x \sim p_x} E_{x' \sim p_x} [|| f(x) - f(x') ||]$ (i.e., no point from $z$ is equal to the center of $z'$).

**Theorem 3.** Denote $r_f(z, z') = c^2 \delta_f(z, z')^2 - | \text{Var}_f[z] - \text{Var}_f[z'] |$.

With probability $1 - \delta$, the coarse-to-fine error is at most:

$$L_{\gamma, f}(z) \leq \frac{\sigma_f(z)}{\sqrt{r_f(z, z')}} - 2 \log \gamma + O\left( \frac{d \log (d/\delta)}{m_z \wedge m_{z'}} \right)^{1/4}$$

under the boundary condition that $r_f(z, z') - 2 \log \gamma \geq 16 \sqrt{2 \log (d/\delta) / m_z \wedge m_{z'}} + 2 \log (d/\delta) / m_z$.

The generalization error depends on the sampling error, $\gamma$, and three quantities intrinsic to the distribution of $f(x)$:

- $s_f(y)$: the bound scales inversely in $s_f(y)$; points in a class must be spread out in order to be subclasses to be distinguishable. Corollary 1 and empirical measurements (Figure 2) suggest that spread is non-zero even when $L_{\text{spread}}$ with $\alpha$ set properly. Note that under SupCon, $s_f(y)$ is asymptotically equal to 0 and this bound is vacuous (refer to Theorem 1 for SupCon’s generalization error).

- $\sigma_f(z)$: the bound scales in $\sigma_f(z)$, confirming that spread alone is insufficient. Subclasses also need to be clustered tightly to achieve good transfer performance.

- $| \text{Var}_f[z] - \text{Var}_f[z'] |$: distinguishing $z$ versus $z'$ may be difficult when only one subclass is clustered. When both $\sigma_f(z)$ and $\sigma_f(z')$ are small, this quantity is negligible.

Altogether, the generalization error scales in $\frac{\sigma_f(z)}{s_f(y)}$. Therefore, in addition to having sufficient spread $s_f(y)$, it is critical that $\sigma_f(z)$ is bounded. We thus explore techniques for inducing an inductive bias that can control this quantity.

### 4.3. Techniques for Inducing Subclass Clustering

We analyze three mechanisms on $f$ for inducing an inductive bias that can cluster subclasses: a constrained encoder, a class-conditional autoencoder, and data augmentations. These three mechanisms use Lipschitzness assumptions of varying strength to bound $\sigma_f(z)$. We assume that subclasses are “clustered” in input space; i.e. there exists some $\sigma_z$ such that $E_{x, x' \sim p_z} || x - x' || \leq \sigma_z$, and we study how these mechanisms on $f$ allow us to control $\sigma_f(z)$ in terms of $\sigma_z$. For each mechanism, we show that $\sigma_f(z) \sim K \sigma_z$ for some particular Lipschitzness constant $K$. The lower the Lipschitzness constant, the better each mechanism can induce subclass clustering.

We summarize the assumptions of each mechanism in Section 4.3.4 and report empirical estimates of Lipschitz constants in Table 1. Section 4.3.4 also reports estimates of $\frac{\sigma_f(z)}{s_f(y)}$, the ratio that governs the generalization error in Theorem 3, showing how these mechanisms impact this quantity.

#### 4.3.1. Lipschitz Encoder

One common method for incorporating inductive bias is to suppose that the class of encoders $F$ is Lipschitz smooth. We show that assuming $F$ to be the class of $K_L$--Lipschitz encoders can explain subclass clustering of representations.

**Lemma 1.** Let $F_{K_L}$ be the class of $K_L$--Lipschitz encoders. Then for any $f_{K_L} \in F_{K_L}$, $\sigma_{f_{K_L}}(z) \leq K_L \sigma_z$.

Lipschitzness with a sufficiently low constant $K_L$ is realistic for simple function classes, such as MLPs with bounded norms. However, modern deep networks are not Lipschitz, as they are powerful enough to memorize random noise (Zhang et al., 2016). In Table 1 we confirm that the Lipschitz constant estimated empirically from our model’s encoder on real data is relatively high. Therefore, since encoders with deep architectures are not Lipschitz, we consider other more realistic setups that can encourage subclass clustering.

#### 4.3.2. Class-Conditional Autoencoder

To encourage embeddings to preserve properties of the input space without assuming Lipschitzness over the encoder, we propose concatenating embeddings from separate “class-conditional” autoencoders, each consisting of an encoder $f_{AE} \in F_{AE}$ and a decoder $g \in G$, to the embeddings learned from $L_{\text{spread}}$. An autoencoder for class $y$ aims to minimize the class reconstruction loss $L_{AE}(D_y) = \frac{1}{m_y} \sum_{x \in D_y} || g(f_{AE}(x)) - x ||^2$. These $K$ per-class autoencoders thus intuitively learn distinct classes within classes.
Define a notion of Rademacher complexity \( \mathcal{R}_N^p(\mathcal{F}_1, \mathcal{F}_2) = \mathbb{E}_\sigma \left[ \sup_{f_1, f_2 \in \mathcal{F}_1, \mathcal{F}_2} \frac{1}{n} \sum_{i=1}^n \sigma_i \| f_1(x_i) - f_2(x_i) \|^p \right] \) for Rademacher random variables \( \sigma = \{ \sigma_1, \ldots, \sigma_n \} \).

**Lemma 2.** For any \( g \in \mathcal{G} \), suppose there exists a \( K_g > 0 \) such that \( g \) is "reverse Lipschitz", satisfying \( \| f_{AE}(x) - f_{AE}(x') \| \leq K_g \| g(f_{AE}(x)) - g(f_{AE}(x')) \| \), and there exists finite \( b \) such that the reconstruction loss satisfies \( \max_x \| g(f_{AE}(x)) - x \|^2 \leq b \).

Then with probability at least \( 1 - \delta \),

\[
\sigma_{f_{AE}}(z) \leq \frac{2K_g}{p(z|y)} \left( L_{AE}(D_y) + 2\mathcal{R}_{n_y}^{2}(G \circ f_{AE} \circ \text{id}_X) \right)^{1/2} + K_g \sigma_z,
\]

(5)

where \( \text{id}_X \) is the identity function on \( X \), and \( p(z|y) = \frac{p(z)}{p(y)} \) is the probability that \( x \) drawn from \( p(y) \) has label \( z \).

There are no explicit assumptions on \( f_{AE} \); instead, a condition on the decoder is used for clustering subclasses. In Appendix C.2, we show that for an autoencoder trained on \( D \) instead of \( D_y \), \( p(z|y) \) replaces \( p(z|y) \), and \( n \) replaces \( n_y \). That is, while a general autoencoder is learned on more data, individual subclasses comprise a smaller proportion of the data and thus could be harder to learn meaningful representations of. This suggests that \( \sigma_{f_{AE}}(z) \) is roughly a constant factor larger with a general autoencoder when \( n_y \) and \( n \) are both large, and thus a class-conditional autoencoder can better cluster subclasses.

**4.3.3. DATA AUGMENTATION**

Another way of inducing inductive bias for subclass clustering is data augmentations, which we use in \( L_{spread} \) and which play a prominent role in contrastive learning overall. Define \( \mathcal{A} : \mathcal{X} \to \mathcal{X} \) as the function class of augmentations, and \( \mathcal{F}_{aug} \) as the class of encoders.

**Lemma 3.** For \( a \in \mathcal{A} \) and any \( x, x' \in \mathcal{X} \), suppose that \( f_{aug} \in \mathcal{F}_{aug} \) satisfies \( \| f_{aug}(a(x)) - f_{aug}(a(x')) \| \leq K_{aug} \| a(x) - a(x') \| \) for some \( K_{aug} \) and that \( f(a(x)) = f(x) \) for \( x \in D \). Denote \( \sigma_{aug} = \mathbb{E}_{x, x' \sim p_x} \| a(x) - a(x') \| \). Then with probability at least \( 1 - \delta \),

\[
\sigma_{f_{aug}}(z) \leq \frac{2}{p(z)} \left( 2\mathcal{R}_{1}^{1}(\mathcal{F}_{aug} \circ \mathcal{F}_{aug} \circ \mathcal{A}) + \sqrt{2\log(1/\delta)/n} \right) + K_{aug} \sigma_{aug}.
\]

Table 1. Three mechanisms for inducing bias and empirical measurements of their associated Lipschitzness constants. Higher \( K \) is a worse Lipschitzness constant, which suggests the assumptions are less realistic.

<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Assumptions</th>
<th>Lipschitzness Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder (Constrained)</td>
<td>Lipschitz ( f )</td>
<td>( K_L = 0.058 )</td>
</tr>
<tr>
<td>Autoencoder</td>
<td>Decoder ( g ) reverse Lipschitz</td>
<td>( K_g = 0.041 )</td>
</tr>
<tr>
<td>Augmentations</td>
<td>Lipschitz ( f ) on augmentations</td>
<td>( K_{aug} = 0.040 )</td>
</tr>
</tbody>
</table>

Our result assumes Lipschitzness only on the augmentations, which is consistent with literature such as Dao et al. (2019), and that the model can align augmented and original training data pairs. \( \sigma_f(z) \) scales with how close augmentations of points within a subclass are, \( \sigma_{aug} \). This quantity can actually be less than \( \sigma_z \) under assumptions in prior work on characterizing augmentations (Huang et al., 2021), which results in tighter embedding clusters. Our result can also explain why prior works (Islam et al., 2021) observe that modified losses (that include augmentations) result in better transfer. The empirical findings in Figure 1 (right), where the subclass embedding visualization are \( L_{spread} \) with and without augmentations, support this result.

**4.3.4. OVERALL TAKEAWAYS**

Our results from Lemmas 1, 2, and 3 show that \( \sigma_f(z) \), which is critical for transfer performance as demonstrated in Theorem 3, can be controlled. Table 1 summarizes our results on how a standard encoder, an autoencoder, and data augmentations can encourage subclass clustering under various assumptions. We report empirical measures of \( K_L, K_g \) and \( K_{aug} \) on real datasets, and find that the autoencoder and data augmentation assumptions are more realistic (lower values of \( K \)).

Figure 2 demonstrates these effects on real data; apparent clusters begin forming under \( L_{spread} \) (which is trained with data augmentation). We also measure the ratio \( \sigma_f(z) \) and find that it can range as high as 1.94 for SupCon. For \( L_{spread} \) with augmentations and the autoencoder, the maximum values are 1.05 and 1.03, respectively—suggesting that these modifications control subclass clustering better, and should result in better coarse-to-fine transfer.

**5. Experiments**

In this section, we evaluate how well adding a class-conditional InfoNCE loss and a class-conditional autoencoder improves the representations produced by supervised contrastive learning. We call our overall method THANOS. This section is primarily designed to evaluate two claims:

- We use coarse-to-fine transfer learning to evaluate how well the representations maintain subclass information. THANOS achieves 11.1 lift on average across five datasets.
We use coarse versions of CIFAR10, CIFAR100, MNIST, and TinyImageNet to study coarse-to-fine transfer. We use Waterbirds, ISIC and CelebA for robustness (Sagawa et al., 2019; Codella et al., 2019; Liu et al., 2015; Sohoni et al., 2020).

Datasets Table 2 lists the datasets we use in our evaluation. We use coarse versions of CIFAR10, CIFAR100, MNIST, and TinyImageNet to study coarse-to-fine transfer. We use Waterbirds, ISIC and CelebA for robustness (Sagawa et al., 2019; Codella et al., 2019; Liu et al., 2015; Sohoni et al., 2020).

Coarse-to-Fine Transfer We use coarse-to-fine transfer learning to isolate how well representations separate subclasses in an ideal setting. In coarse-to-fine transfer, we train models on coarse labels, freeze the weights, and then train a linear probe over the final layer on fine labels. Note that this setting is more challenging than the self-supervised setting, since it requires maintaining high performance on the coarse classes while also being transferrable to the fine classes. We focus on transfer numbers in this section, but Table 6 in the Appendix presents results on coarse accuracy.

For the autoencoder experiments, we train an autoencoder separately and concatenate its embedding layer with the contrastive embedding for the linear probe. We jointly optimize all contrastive losses and the class-conditional autoencoder with a cross-entropy loss head. We train all models with dropout as well as label smoothing on the cross-entropy loss heads.

We report four variants of THANOS: the class-conditional autoencoder on its own, SupCon modified with a class-conditional InfoNCE loss, SupCon modified with a class-conditional autoencoder, and SupCon with both modifications. We report 3 baselines from previous work on the transferability of SupCon (Islam et al., 2021): SupCon, SupCon plus an InfoNCE loss, and the InfoNCE loss on its own.

THANOS significantly outperforms SupCon on coarse-to-fine transfer learning—by an average of 11.1 points across all tasks. 7.3 points can be attributed to the class-conditional InfoNCE loss on average, but mileage varies between tasks (25.3 points of lift for CIFAR10, vs. 2.6 for CIFAR100). The difference is the number of coarse classes: CIFAR10 only has two coarse classes, whereas CIFAR100 has 20. Fewer coarse classes makes it easier to achieve class collapse, so spread is more necessary. Finally, we also note that combining the autoencoder with the other components outperforms using the autoencoder on its own, by 2.7 points on average. This suggests that each component is helpful.

Worst-Group Robustness We use robustness to measure how well THANOS can recover hidden subgroups in an unsupervised setting. For these models, we train contrastive losses on their own. We follow the methodology from Sohoni et al. (2020). We first train a model with class labels. We then cluster the embeddings to produce pseudolabels for subclasses, which we use as input to the GroupDRO algorithm to optimize worst-group robustness (Sagawa et al., 2019). Our primary evaluation metric is robustness, but we also evaluate a subgroup recovery metric since prior work has suggested that it is important for robustness. Subgroup recovery also acts as a proxy for unsupervised group recovery. We compare subgroup recovery against SupCon and Sohoni et al. (2020). We compare worst-group robustness against Sohoni et al. (2020) and JTT (Liu et al., 2021), as well as using subgroup labels from SupCon. We also report the performance of GroupDRO with ground-truth subclass labels.

Table 4 shows the results. THANOS outperforms both SupCon and Sohoni et al. (2020) on subgroup recovery. THANOS further achieves state-of-the-art worst-group robustness, outperforming JTT by 4.7 points and Sohoni et al. (2020) by 11.7 points on average—and setting state-of-the-art on CelebA by 11.5 points. Surprisingly, THANOS can even outperform GroupDRO with ground-truth subgroup labels in two cases.

\[ \text{Table 2. Summary of the datasets we use for evaluation.} \]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$K_{\text{coarse}}$</th>
<th>$K_{\text{fine}}$</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>2</td>
<td>10</td>
<td>Coarse labels are animal vs. vehicle</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>20</td>
<td>100</td>
<td>Standard coarse labels</td>
</tr>
<tr>
<td>CIFAR100-U</td>
<td>20</td>
<td>100</td>
<td>CIFAR100, imbalanced fine classes</td>
</tr>
<tr>
<td>MNIST</td>
<td>2</td>
<td>10</td>
<td>Coarse labels are $&lt;5$ and $\geq 5$</td>
</tr>
<tr>
<td>TinyImageNet</td>
<td>67</td>
<td>200</td>
<td>Coarse labels from ImageNet hierarchy</td>
</tr>
<tr>
<td>Waterbirds</td>
<td>2</td>
<td>3</td>
<td>Bird images (Sagawa et al., 2019)</td>
</tr>
<tr>
<td>ISIC</td>
<td>2</td>
<td>3</td>
<td>Skin lesions (Codella et al., 2019)</td>
</tr>
<tr>
<td>CelebA</td>
<td>2</td>
<td>3</td>
<td>Celebrity faces (Liu et al., 2015)</td>
</tr>
</tbody>
</table>

\[ \text{Notes} \]

\[ \text{We also present ablations. Additional experiments on overall model quality, additional baselines, and additional datasets are in Appendix G. Although we focus on coarse-to-fine transfer and robustness here, we note that our method also produces lift on overall model quality.} \]

THANOS Method We summarize the THANOS method.\footnote{Our code is available at \url{https://github.com/HazyResearch/thanos-code/}.} THANOS consists of adding a class-conditional InfoNCE loss and a class-conditional autoencoder to the supervised contrastive loss with standard data augmentations used in Chen et al. (2020a). We implement the former by training an encoder with $L_{\text{spread}}$. To implement the latter, we train a single autoencoder with a joint MSE reconstruction loss and a cross entropy loss. We then concatenate the autoencoder representation to the representation of the encoder trained with $L_{\text{spread}}$. Details on architectures and hyperparameters in Appendix F.
Table 4. Unsupervised subclass recovery (top, F1), and worst-group performance (AUROC for ISIC, Acc for others). Best in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Group Labels</th>
<th>Waterbirds</th>
<th>ISIC</th>
<th>CelebA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baselines</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InfoNCE (Chen et al., 2020a)</td>
<td></td>
<td>77.6 ± 0.1</td>
<td>60.5 ± 0.1</td>
<td>56.4 ± 0.3</td>
</tr>
<tr>
<td>SupCon (Khosla et al., 2020)</td>
<td></td>
<td>51.8 ± 1.2</td>
<td>56.1 ± 0.1</td>
<td>49.8 ± 0.3</td>
</tr>
<tr>
<td>SupCon + InfoNCE (Islam et al., 2021)</td>
<td></td>
<td>77.6 ± 0.1</td>
<td>55.7 ± 0.1</td>
<td>48.0 ± 0.2</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cAuto</td>
<td></td>
<td>71.4 ± 0.1</td>
<td>62.9 ± 0.1</td>
<td>58.7 ± 0.5</td>
</tr>
<tr>
<td>SupCon + cNCE (Lspread)</td>
<td></td>
<td>71.7 ± 0.1</td>
<td>58.7 ± 0.2</td>
<td>53.5 ± 0.4</td>
</tr>
<tr>
<td>SupCon + cAuto</td>
<td></td>
<td>71.7 ± 0.1</td>
<td>63.8 ± 0.6</td>
<td>59.8 ± 0.3</td>
</tr>
<tr>
<td>SupCon + cNCE + cAuto (THANOS)</td>
<td></td>
<td><strong>79.1 ± 0.2</strong></td>
<td><strong>65.0 ± 0.2</strong></td>
<td>59.7 ± 0.3</td>
</tr>
</tbody>
</table>

Subgroup recovery and worst-group robustness are correlated but not causal: (Sohoni et al., 2020) observed inconsistencies between them, and so do we (i.e., our method outperforms GroupDRO, an approach with “perfect” subgroup labels). This phenomenon deserves further exploration.

Ablations We summarize two ablations (Appendix G.5). First, we validate Lemma 2 and find that using a generic autoencoder underperforms a class-conditional autoencoder by 30.0 points on average—and furthermore does not improve the performance of SupCon as well (2.0 points of lift compared to 11.0 points). Second, we validate Lemma 3 and confirm that data augmentation is crucial; removing data augmentation degrades performance by 35.4 points.

6. Related Work and Discussion

We present an abbreviated related work. A full treatment can be found in Appendix A. Our theoretical work relates to theory on the geometry of contrastive learning (Wang & Isola, 2020; Graf et al., 2021; Robinson et al., 2020; Zimmermann et al., 2021), collapsed representations (Galanti et al., 2021; Jing et al., 2021), autoencoders (Epstein & Meir, 2019; Le et al., 2018), data augmentation (HaoChen et al., 2021; Guo et al., 2018; Abavisi et al., 2020), and robustness (Sohoni et al., 2021). Our use of \( L_{\text{spread}} \) and an autoencoder draws from a wave of empirical work on contrastive learning (Chen et al., 2020a; Khosla et al., 2020), and its properties (Islam et al., 2021; Bukchin et al., 2021).

In aggregate, we study how to improve the quality of representations trained with supervised contrastive learning. We identify controlling spread and inducing subclass clustering as two key challenges and show how two modifications to supervised contrastive learning improve transfer and robustness.

Authors’ Note The first two authors contributed equally. Co-first authors can prioritize their names when adding this paper’s reference to their resumes.

Acknowledgments

We thank Beidi Chen, Tri Dao, Karan Goel, and Albert Gu for their helpful comments on early drafts of this paper. We gratefully acknowledge the support of NIH under No. U54EB020405 (Mobilize), NSF under Nos. CCF1763315 (Beyond Sparsity), CCF1563078 (Volume to Velocity), and 1937301 (RTML); ONR under No. N000141712266 (Unifying Weak Supervision); ONR N00014-20-1-2480: Understanding and Applying Non-Euclidean Geometry in Machine Learning; N000142012275 (NEPTUNE); the Moore Foundation, NXP, Xilinx, LETI-CEA, Intel, IBM, Microsoft, NEC, Toshiba, TSMC, ARM, Hitachi, BASF, Accenture, Ericsson, Qualcomm, Analog Devices, the Okawa Foundation, American Family Insurance, Google Cloud, Salesforce, Total, the HAI-GCP Cloud Credits for Research program, the Stanford Data Science Initiative (SDSI), Department of Defense (DoD) through the National Defense Science and Engineering Graduate Fellowship (NDSEG) Program, and members of the Stanford DAWN project: Facebook, Google, and VMWare. The Mobilize Center is a Biomedical Technology Resource Center, funded by the NIH National Institute of Biomedical Imaging and Bioengineering through Grant P41EB027060. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views, policies, or endorsements, either expressed or implied, of NIH, ONR, or the U.S. Government.
References


Huang, W., Yi, M., and Zhao, X. Towards the generalization of contrastive self-supervised learning, 2021.


A. Related Work

We presented an extended treatment of related work.

From work in contrastive learning, we take inspiration from Arora et al. (2019), who use a latent classes view to study self-supervised contrastive learning. Similarly, Zimmermann et al. (2021) considers how minimizing the InfoNCE loss recovers a latent data generating model. Recent work has also analyzed contrastive learning from the information-theoretic perspective (Oord et al., 2018; Tian et al., 2020; Tsai et al., 2020), but does not fully explain practical behavior (Tschannen et al., 2020). On the geometric side, we are inspired by the theoretical tools from Wang & Isola (2020) and Graf et al. (2021), who study representations on the hypersphere along with Robinson et al. (2020). There has been work studying other notions of collapsed representations. Jing et al. (2021) examines dimension collapse in contrastive learning, which occurs when the learned representations span a subspace of the representation space. Our definition of class collapse can also be viewed as Neural Collapse (Papyan et al., 2020), which started as an empirical observation about when models are trained beyond 0 training error using cross-entropy or MSE loss (Lu & Steinerberger, 2020; Han et al., 2021). Recent works on neural collapse have studied the transferrability of collapsed representations (Galanti et al., 2021), and Hui et al. (2022); Kothapalli et al. (2022) have identified its limitations in this setting. We offer another perspective on the relationship between collapse and embedding quality, and offer techniques to mitigate the effects of collapse in transfer learning.

Our work builds on the recent wave of empirical interest in contrastive learning (Chen et al., 2020a; He et al., 2019; Chen et al., 2020b; Goyal et al., 2021; Caron et al., 2020) and supervised contrastive learning (Khosla et al., 2020). There has also been empirical work analyzing the transfer performance of contrastive representations and the role of intra-class variability in transfer learning. Islam et al. (2021) find that combining supervised and self-supervised contrastive loss improves transfer learning performance, and they hypothesize that this is due to both inter-class separation and intra-class variability. Bukchin et al. (2021) find that combining cross entropy and a class-conditional self-supervised contrastive loss improves coarse-to-fine transfer, also motivated by preserving intra-class variability.

Our use of $L_{spread}$ and a class-conditional autoencoder arises from similar motivations to losses proposed in these works, and we further theoretically study their implications for spread. Our theoretical analysis of autoencoders draws from previous work (Epstein & Meir, 2019; Le et al., 2018). Our study of data augmentation similarly builds on recent theoretical analysis of the role of data augmentation in contrastive learning (HaoChen et al., 2021; Huang et al., 2021) and clustering (Guo et al., 2018; Abavisani et al., 2020). Our treatment of subclasses is strongly inspired by Sohoni et al. (2020) and Oakden-Rayner et al. (2020), who document empirical consequences of hidden strata. We are inspired by empirical work that has demonstrated that detecting subclasses can be important for performance (Hoffmann et al., 2001; d’Eon et al., 2021) and robustness (Duchi et al., 2020; Sagawa et al., 2019; Goel et al., 2020; Liu et al., 2021).

B. Glossary

The glossary is given in Table 5 below.
Table 5. Glossary of variables and symbols used in this paper.
C. Proofs

C.1. Proofs for Section 3

Theorem 1. For $\gamma$ where $\log \gamma \geq 8 \max\{R_{n_z}(F), R_{n_x}(F)\}$, SupCon’s coarse-to-fine error is at least

$$L_{\gamma,f_{SC}}(z) \geq 1 - \delta(n_z,F,\gamma) - \delta(n_x,F,\gamma) - \xi(m_z \wedge m_x, \gamma),$$

where $\delta(n_x,F,\gamma) = d \exp \left( - \frac{n}{32d^2} (\log \gamma - 8R_{n_x}(F))^2 \right)$ bounds generalization error of $f_{SC}$ and $\xi(m_z \wedge m_x, \gamma) = 4d \exp \left( - \frac{(m_z \wedge m_x) \log 2 \gamma}{32d} \right)$ bounds the noise from $D_z$.

Proof. $L_{\gamma,f}(z) = \Pr_{x \sim P_s}(\hat{\rho}(z) | f(x)) \leq \gamma \hat{\rho}(z' | f(x))$, and by definition of the linear softmax classifier, we have that

$$L_{\gamma,f}(z) = \Pr_{x \sim P_s} \left( \exp(f(x)^\top W_z) \leq \gamma \exp(f(x)^\top W_{z'}) \right) = \Pr_{x \sim P_s} \left( f(x)^\top (W_z - W_{z'}) \leq \log \gamma \right).$$

(6)

To lower bound this quantity, we focus on upper bounding $f(x)^\top (W_z - W_{z'})$. We can bound $f_{SC}(x)^\top (W_z - W_{z'}) \leq \|W_z - W_{z'}\| \leq \|\mathbb{E}_{x \sim P_s}[f_{SC}(x) - f_{SC}(x')]\| + \xi + \xi'$, where $\xi = \left\| \frac{1}{n_z} \sum_{z \in D_z} (f_{SC}(x) - \mathbb{E}_{x \sim P_s}[f_{SC}(x)]) \right\|$. This can be bounded via standard concentration inequalities and $\xi'$ is similarly constructed.

Because SupCon yields collapsed training embeddings within any class, we know that $f_{SC}(x) = f_{SC}(x')$ for $x,x' \in D$ where $h_s(x) = z$ and $h_s(x') = z'$. Therefore, it holds that

$$\|\mathbb{E}_{x \sim P_s}[f_{SC}(x)] - \mathbb{E}_{x' \sim P_s}[f_{SC}(x')]\| \leq \|\mathbb{E}_{x \sim P_s}[f_{SC}(x)] - \mathbb{E}_{x \sim P_s}[f_{SC}(x')]\| + \xi + \xi'.$$

Define $\epsilon(z,D,F) = \sup_{f \in F} \|\mathbb{E}_{x \sim P_s}[f(x)] - \frac{1}{n_z} \sum_{x \in D_z} f(x)\|$ and similarly $\epsilon(z',D,F)$. Therefore, our loss in (6) satisfies

$$L_{\gamma,f_{SC}}(z) \geq \Pr(\epsilon(z,D,F) + \epsilon(z',D,F) + \xi + \xi' \leq \log \gamma) \geq \Pr(\epsilon(z,D,F) \leq \log \gamma / 4) \Pr(\epsilon(z',D,F) \leq \log \gamma / 4) \Pr(\xi \leq \log \gamma / 4) \Pr(\xi' \leq \log \gamma / 4),$$

(7)

where independence comes from the fact that data is i.i.d. sampled for each subclass and each $D$ and $D_s$, and that we are taking the supremum over $F$. Next, we bound $\epsilon(z,D,F)$. Since $f(x) \leq 1$, we have that by Lemma 6 that with probability $1 - \delta$,

$$\epsilon(z,D,F) \leq 2R_{n_z}(F) + d \sqrt{\frac{2 \log(d/\delta)}{n_z}}.$$ 

Setting $\epsilon := 2R_{n_z}(F) + d \sqrt{\frac{2 \log(d/\delta)}{n_z}}$, we can write $\delta = d \exp \left( - \frac{n}{32d^2} (\epsilon - 2R_{n_z}(F))^2 \right)$. Therefore, for $\log \gamma \geq 8R_{n_z}(F)$, we have that

$$\Pr(\epsilon(z,D,F) \leq \log \gamma / 4) \geq 1 - d \exp \left( - \frac{n_z}{32d^2} (\log \gamma - 8R_{n_z}(F))^2 \right).$$

(8)

Next, we bound $\xi_z$. We can write

$$\left\| \frac{1}{n_z} \sum_{x \in D_z} f_{SC}(x) - \mathbb{E}_{x \sim P_s}[f_{SC}(x)] \right\|^2 \leq \left( \sum_{j=1}^{d} \left( \frac{1}{m_z} \sum_{x \in D_z} f_{SC}(x)[j] - \mathbb{E}_{x \sim P_s}[f_{SC}(x)[j]] \right)^2 \right)^{1/2},$$

where $f_{SC}(x)[j]$ is the $j$th element of $f_{SC}(x)$. Using Hoeffding’s inequality, we have that $\Pr(\left\| \frac{1}{m_z} \sum_{x \in D_z} f_{SC}(x)[j] - \mathbb{E}_{x \sim P_s}[f_{SC}(x)] \right\| \leq \delta) \geq 1 - 2e^{-2\delta^2 m_z}$.
We can thus write

\[\Pr \left( \sum_{j=1}^{d} \left( \frac{1}{m_z} \sum_{x \in D_{n,z}} f_{SC}(x)[j] - \mathbb{E}_{x \sim p_z}[f_{SC}(x)[j]] \right)^2 \leq \epsilon \right) \geq \Pr \left( \sum_{j=1}^{d} \left( \frac{1}{m_z} \sum_{x \in D_{n,z}} f_{SC}(x)[j] - \mathbb{E}_{x \sim p_z}[f_{SC}(x)[j]] \right)^2 \leq \epsilon \right) \geq \left( 1 - 2\exp \left( - \frac{m_z \epsilon}{2} \right) \right)^{d} \geq 1 - 2d\exp \left( - \frac{m_z \epsilon}{2} \right).
\]

That is, \( \Pr(\xi \leq \sqrt{d}\epsilon) \geq 1 - 2d\exp \left( - \frac{m_z \epsilon}{2} \right) \).

Setting \( \log \frac{1}{d} = \sqrt{d} \epsilon \) gives us \( \Pr(\xi \leq \log \frac{1}{d}) \geq 1 - 2d\exp \left( - \frac{m_z \log^2 \gamma / 16d}{2} \right) = 1 - 2d\exp \left( - \frac{m_z \log^2 \gamma}{32d} \right) \).

We put this expression and (8) back into (7) and use the fact that \((1 - \delta_1)(1 - \delta_2) \geq 1 - \delta_1 - \delta_2\) for any \(\delta_1, \delta_2 > 0\). Therefore, we have for SupCon,

\[
L_{f,SC}(\gamma) \geq \frac{1}{d}d\exp \left( - \frac{n_z}{32d^2} (\log \gamma - 8\mathbb{R}_{n_z}(\mathcal{F}))^2 \right) - \exp \left( - \frac{n_z}{32d^2} (\log \gamma - 8\mathbb{R}_{n_z}(\mathcal{F}))^2 \right) - 2d\exp \left( - \frac{m_z \log^2 \gamma}{32d} \right) - 2d\exp \left( - \frac{m_z \log^2 \gamma}{32d} \right) \geq \frac{1}{d}d\exp \left( - \frac{n_z}{32d^2} (\log \gamma - 8\mathbb{R}_{n_z}(\mathcal{F}))^2 \right) - 4d\exp \left( - \frac{(m_z + m_z') \log^2 \gamma}{32d} \right) = 1 - \delta(n_z, \mathcal{F}, \gamma) - \delta(n_z', \mathcal{F}, \gamma) - \xi(m_z, m_z', \gamma).
\]

\[\Box\]

**Derivation of \(L_{\text{spread}}(f, \alpha)\)**: We explain how we arrive at the asymptotic form of \(L_{\text{spread}}\). Let \(L_{\text{spread}}(f, n^+, n^-)\) be the population-level version of \(\hat{L}_{\text{spread}}(f, B)\), where \(n^+, n^-\) are the number of negatives in the denominators of \(L_{\text{cNCE}}\) and \(L_{\text{sup}}\), respectively.

\[
L_{\text{spread}}(f, n^+, n^-) = (1 - \alpha)L_{\text{sup}}(f, n^-) + \alpha L_{\text{cNCE}}(f, n^+) \tag{9}
\]

\[
L_{\text{sup}}(f, n^-) = -\mathbb{E} \left[ \log \frac{\sigma_f(x, x^+)}{\sigma_f(x, x^-) + \sum_{i=1}^{n^-} \sigma_f(x, x_i^-)} \right] \tag{10}
\]

\[
L_{\text{cNCE}}(f, n^+) = -\mathbb{E} \left[ \log \frac{\sigma_f(x, a(x))}{\sigma_f(x, a(x)) + \sum_{i=1}^{n^+} \sigma_f(x, x_i^+)} \right] \tag{11}
\]

We now demonstrate how minimizing \(\lim_{n^+, n^- \to \infty} L_{\text{spread}}(f, n^+, n^-)\) is equivalently to minimizing \(L_{\text{spread}}(f, \alpha)\). In \(L_{\text{sup}}\), we divide the numerator and denominator by \(n^-\), and in \(L_{\text{cNCE}}\) we divide the numerator and denominator by \(n^+\):

\[
L_{\text{sup}}(f, n^-) = \mathbb{E} \left[ \log \frac{\exp(\sigma_f(x, x^+))}{\frac{1}{n^-} \exp(\sigma_f(x, x^+)) + \frac{1}{n^-} \sum_{i=1}^{n^-} \exp(\sigma_f(x, x_i^-))} \right] + \log n^-
\]

\[
L_{\text{cNCE}}(f, n^+) = \mathbb{E} \left[ \log \frac{\exp(\sigma_f(x, a(x)))}{\frac{1}{n^+} \exp(\sigma_f(x, a(x))) + \frac{1}{n^+} \sum_{i=1}^{n^+} \exp(\sigma_f(x, x_i^+))} \right] + \log n^+
\]

We can thus write \(L_{\text{spread}}(f, n^+, n^-)\) as

\[
L_{\text{spread}}(f, n^+, n^-) = \alpha \log n^- + (1 - \alpha) \log n^+ - \alpha \mathbb{E}[\sigma_f(x, x^+)] - (1 - \alpha) \mathbb{E}[\sigma_f(x, a(x))]
\]

\[
+ \alpha \mathbb{E} \left[ \log \left( \frac{1}{n^-} \exp(\sigma_f(x, x^+)) + \frac{1}{n^-} \sum_{i=1}^{n^-} \exp(\sigma_f(x, x_i^-)) \right) \right]
\]

\[
+ (1 - \alpha) \mathbb{E} \left[ \log \left( \frac{1}{n^+} \exp(\sigma_f(x, a(x))) + \frac{1}{n^+} \sum_{i=1}^{n^+} \exp(\sigma_f(x, x_i^+)) \right) \right].
\]
Perfectly Balanced: Improving Transfer and Robustness of Supervised Contrastive Learning

Taking the limit \( n^+, n^- \to \infty \) yields

\[
\lim_{n^+, n^- \to \infty} L_{\text{spread}}(f, n^+, n^-) = -\alpha \log n^- \quad (1-\alpha) \log n^+ = -\alpha E_{x,x'}[\sigma_f(x,x')] - (1-\alpha) E[\sigma_f(x,a(x))]
\]

\[+ \alpha E_{x,x'}[\log E_{x,x'}[\exp(\sigma_f(x,x'))]] + (1-\alpha) E_x[\log E_{x+[\exp(\sigma_f(x,x'))]]].
\]

Lastly, we use the fact that \( \sigma_f(x,x') = \exp(f(x) - f(x'))/\tau = \exp(-\|f(x) - f(x')\|^2/2\tau), \) since \( f(x), f(x') \in \mathcal{S}^{d-1} \) to get that

\[
\lim_{n^+, n^- \to \infty} L_{\text{spread}}(f, n^+, n^-) = -(1-\alpha) L_{\text{diff}}(f) + (1-\alpha) L_{\text{align}}(f) + (1-\alpha) L_{\text{aug}}(f) + (1-\alpha) L_{\text{name}}(f)
\]

Proposition 1 (Individual losses). \( L_{\text{align}}(f) \) and \( L_{\text{aug}}(f) \) are minimized when \( f(x) = f(x^+) \) and \( f(x) = f(a(x)) \) almost surely, respectively. \( L_{\text{diff}}(\mu) \) is minimized when \( \mu = \delta_v \). \( L_{\text{same}}(\mu) \) is minimized when \( \mu = \sigma_{d-1} \).

Proof. We analyze each term’s optimal measure on the hypersphere.

\( L_{\text{align}}(f), L_{\text{aug}}(f) \) For both \( L_{\text{align}}(f) \) and \( L_{\text{aug}}(f) \), the minimum value of the expression is 0, which is obtained when \( f(x) = f(x^+) \) and \( f(x) = f(a(x)) \) almost surely, respectively.

\( L_{\text{diff}}(f) \) We show in Lemma 5 that the measure that minimizes \( L_{\text{diff}}(f) \) also minimizes \( \log E_{x,x'}[\exp(-\|f(x) - f(x')\|^2/2\tau)] \) (note this is not identical to the approach in Wang & Isola (2020)). We can thus equivalently consider minimizing \( \log E_{x,x'}[\exp(-\|f(x) - f(x')\|^2/2\tau)] \). Note that \( \max_{f(x), f(x') \in \mathcal{S}^{d-1}} \|f(x) - f(x')\| = 2 \), and so \( \inf_{x,x'}[\exp(-\|f(x) - f(x')\|^2/2\tau)] = \exp(-2/\tau) \). For \( \mu = \delta_v \),

\[
E_{x,x'}[\exp(-\|f(x) - f(x')\|^2/2\tau)] = \frac{1}{2} E_{h(x)=0,h(x')=1}[\exp(-\|v_0 - v_1\|^2/2\tau)] + \frac{1}{2} E_{h(x)=1,h(x')=0}[\exp(-\|v_1 - v_0\|^2/2\tau)] = \exp(-2/\tau).
\]

The first equality follows from class balance, and the third equality follows from the definition of the regular simplex. Therefore, \( \mu = \delta_v \) minimizes \( L_{\text{diff}}(f) \).

\( L_{\text{same}}(f) \) For \( L_{\text{same}}(f) \), we can directly use the proof of Theorem 1 in Wang & Isola (2020) to show that the optimal measure that minimizes \( L_{\text{same}}(f) \) also minimizes \( \log E_{x,x'}[\exp(-\|f(x) - f(x')\|^2/2\tau)] \). We equivalently consider minimizing \( \log E_{x,x'}[\exp(-\|f(x) - f(x')\|^2/2\tau)] \). We can write this as

\[
E_{x,x'}[\exp(-\|f(x) - f(x')\|^2/2\tau)] = \frac{1}{2} E_{h(x)=0,h(x')=1}[\exp(-\|f(x) - f(x')\|^2/2\tau)] + \frac{1}{2} E_{h(x)=1,h(x')=0}[\exp(-\|f(x) - f(x')\|^2/2\tau)]
\]

Using the infinite encoder assumption, we can equivalently consider the following minimization problem over the hypersphere, where \( u, u' \in \mathcal{S}^{d-1} \):

\[
\text{minimize } \mu_0, \mu_1, \frac{1}{2} \int \int \exp(-\|u - u'\|^2/2\tau) d\mu_0(u) d\mu_0(u') + \frac{1}{2} \int \int \exp(-\|u - u'\|^2/2\tau) d\mu_1(u) d\mu_1(u')
\]

Each of these integrals can be minimized individually, and the problem becomes equivalent to having both \( \mu_0 \) and \( \mu_1 \) minimize the Gaussian \( \frac{1}{\tau} \)-energy. Using Proposition 4.4.1 and Theorem 6.2.1 of (Borodachov et al., 2019), the optimal solution is \( \mu_0 = \mu_1 = \sigma_{d-1} \), the normalized surface area measure. Therefore, \( \mu = \sigma_{d-1} \).

\[
\text{Theorem 2. Let } c_{r,d} = \frac{2^{d+1} - \sqrt{2} (2^{d+1} - 2 \log W_{1/2}(S^{d-1}))}{3 \pi^{(d-1)/2}} W_{1/2}(S^{d-1}), \text{ where } W_{1/2}(S^{d-1}) \text{ is the Wiener constant of the Gaussian } \frac{1}{\tau} \text{-energy on } S^{d-1}, \text{ which is defined as}
\]

\[
W_{1/2}(S^{d-1}) = \frac{2^{d-2} \Gamma(d/2)}{\sqrt{\pi} \Gamma((d-1)/2)} \int_0^1 \exp\left(-\frac{2u}{\tau}\right) (u(1-u))^{(d-3)/2} du,
\]
where the Gamma function is \( \Gamma(z) = \int_0^\infty x^{z-1}e^{-x}dx \) for \( z > 0 \). Then, when \( \alpha \in (2/3, c_{\tau, d}) \), \( \theta^* = \arcsin \sqrt{\frac{\tau}{2}\log \frac{3\alpha - 1}{3\alpha - 3}} \)
minimizes \( L_{\text{spread}}(\mu_0, \alpha) \) and satisfies \( L_{\text{spread}}(\mu_0^*, \alpha) \leq \min_{\mu \in (\delta_\alpha, \sigma_d + 1)} L_{\text{spread}}(\mu, \alpha) \).

**Proof.** Because the augmentations only play a role in \( L_{\text{aug}}(f) \) and are disjoint, the condition that \( f(x) = f(a(x)) \) a.s. is compatible with any of the other three losses in Proposition 1. Therefore, we focus on analyzing the combined weighted loss \((1-\alpha)L_{\text{align}}(f) + (1-\alpha)L_{\text{diff}}(f) + \alpha L_{\text{same}}(f)\). We restate the loss:

\[
L(\mu, \alpha) = (1-\alpha)\mathbb{E}_x \left[ \log \mathbb{E}_{x'} \left[ \exp \left( -\frac{1}{2\tau} \|f(x) - f(x')\|^2 \right) \right] \right] + \alpha \mathbb{E}_x \left[ \log \mathbb{E}_{x'} \left[ \exp \left( -\frac{1}{2\tau} \|f(x) - f(x')\|^2 \right) \right] \right]
\]

We restate the definition of the measure \( \mu_\theta = \{\mu_{0, \theta}, \mu_{1, \theta}\} \) for \( \theta \in (0, \pi/2] \) that involves “splitting” \( \delta_\alpha \) by angle \( \theta \). Without loss of generality, suppose that \( v_0 = e_1 \) and \( v_1 = -e_1 \), where \( e_1 \in \mathbb{R}^d \) is a standard basis vector \([1, 0, \ldots, 0]\) where all but the first two elements are always 0. For \( v_0 \), we consider the vectors \( v_{0, \theta} = [\cos(\theta), \sin(\theta), \ldots, 0] \) and \( v_{0, -\theta} = [\cos(\theta), -\sin(\theta), \ldots, 0] \). For \( v_1 \), we consider the vectors \( v_{1, \theta} = [-\cos(\theta), -\sin(\theta), \ldots, 0] \) and \( v_{1, -\theta} = [-\cos(\theta), \sin(\theta), \ldots, 0] \). Let \( \mu_{0, \theta} = \frac{1}{2} \delta_{v_{0, \theta}} + \frac{1}{2} \delta_{v_{0, -\theta}} \) and let \( \mu_{1, \theta} = \frac{1}{2} \delta_{v_{1, \theta}} + \frac{1}{2} \delta_{v_{1, -\theta}} \). That is, each class-condition measure is a mixture on two points separated by \( \theta \).

The first step is to show that for some range of \( \alpha, \min_{\theta} L(\mu_\theta, \alpha) < L(\delta_\alpha, \alpha) \). We have that

\[
L(\delta_\alpha, \alpha) = (1-\alpha) \left( \frac{1}{2} \log \exp \left( - \frac{4}{2\tau} \right) + \frac{1}{2} \log \exp \left( - \frac{4}{2\tau} \right) \right) + \alpha \log \exp (0) + (1-\alpha)0 = -\frac{2(1-\alpha)}{\tau}.
\]

and

\[
L(\mu_\theta, \alpha) = (1-\alpha) \log \left( \frac{1}{2} \exp \left( - \frac{4}{2\tau} \right) + \frac{1}{2} \exp \left( - \frac{4\cos^2(\theta)}{2\tau} \right) \right) + \alpha \log \left( \frac{1}{2} + \frac{1}{2} \exp \left( - \frac{4\sin^2(\theta)}{2\tau} \right) \right) + \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{4\sin^2(\theta)}{\tau} + (1-\alpha) \log \left( 1 + \exp \left( \frac{2\sin^2(\theta)}{\tau} \right) \right) + \alpha \log \left( 1 + \exp \left( - \frac{2\sin^2(\theta)}{\tau} \right) \right)
\]

We now compute the derivative \( \frac{\partial L(\mu_\theta, \alpha)}{\partial \theta} \) to find local minima:

\[
\frac{\partial L(\mu_\theta, \alpha)}{\partial \theta} = (1-\alpha) \frac{\exp(2\sin^2(\theta)/\tau) \cdot 4 \sin \theta \cos \theta / \tau}{1 + \exp(2\sin^2(\theta)/\tau)} + \alpha \frac{\exp(-2\sin^2(\theta)/\tau) \cdot 4 \sin \theta \cos \theta / \tau}{1 + \exp(-2\sin^2(\theta)/\tau)} + \frac{(1-\alpha)4 \sin \theta \cos \theta}{\tau}
\]

Note that \( \sin \theta \) and \( \cos \theta \) are positive for \( \theta \in (0, \pi/2] \). Next, for notational simplicity let \( x = \frac{2\sin^2(\theta)}{\tau} \). Then, we can equivalently evaluate

\[
(1-\alpha) \frac{e^x}{1+e^x} - \alpha \frac{e^{-x}}{1+e^{-x}} + \frac{1}{2} = (1-\alpha) \frac{e^x}{1+e^x} - \alpha \frac{1}{1+e^x} + \frac{1-\alpha}{2}
\]

\[
= \frac{e^x - \alpha(1+e^x)}{1+e^x} + \frac{1-\alpha}{2} = \frac{e^x}{1+e^x} + \frac{1}{2} \cdot \frac{3\alpha}{2}.
\]

Setting this equal to 0, we get that \( \alpha = \frac{3e^x + 1}{2e^x + 3} \) and \( x = \log \frac{3e^x - 1}{3e^x + 3} \). Since \( x \in (0, 2/\tau] \), we have that if \( \alpha \in \left( \frac{2}{3}, \frac{3\exp(2\tau)/3+1}{3\exp(2\tau)/3+3} \right) \), there exists a local optima over \( \theta \in (0, \pi/2] \).

Moreover, we observe that when \( \alpha \leq 2/3 \), we have that \( \frac{e^x}{1+e^x} + \frac{1}{2} \geq \frac{3\alpha}{2} \geq 0 \), which means that \( L(\mu_\theta, \alpha) \) increases in \( \theta \) for \( \alpha \leq 2/3 \). Therefore, when \( \alpha \leq 2/3 \), class collapse is always better no matter the angle, and \( L(\delta_\alpha, \alpha) \leq \min_{\theta} L(\mu_\theta, \alpha) \).

Next, we consider when \( \alpha \geq \frac{3\exp(2\tau)/3+1}{3\exp(2\tau)/3+3} \). In this case, \( \frac{e^x}{1+e^x} + \frac{1}{2} \geq \frac{3\alpha}{2} \), so \( L(\mu_\theta, \alpha) \) is decreasing in \( \theta \). This means that any nonzero \( \theta \) in this setting is going to result in a smaller loss than the class-collapsed loss.
Lastly, we consider the intermediate case of \( \alpha \in \left( \frac{2}{3}, \frac{3\exp(2/\tau)+1}{3\exp(2/\tau)+3} \right) \). Plugging back in \( x = \log \frac{3\alpha-1}{3\alpha} \) back into \( L(\mu_\theta, \alpha) \) in (16), we have
\[
L(\mu_\theta, \alpha) = -\log 2 - \frac{2(1-\alpha)}{\tau} + (1-\alpha) \log \left( 1 + \frac{3\alpha-1}{3-3\alpha} \right) + \alpha \log \left( 1 + \frac{3\alpha-1}{3-3\alpha} \right) - \frac{1-\alpha}{2} \log \frac{3\alpha-1}{3-3\alpha} = -\log 2 - \frac{2(1-\alpha)}{\tau} + (1-\alpha) \log \frac{3\alpha-1}{3-3\alpha} + \alpha \log \frac{3\alpha-1}{3-3\alpha} - \frac{1-\alpha}{2} \log \frac{3\alpha-1}{3-3\alpha}.
\] (17)

Note that \( (3-3\alpha) \log (3-3\alpha) + (3\alpha-1) \log (3\alpha-1) \) equals 0 at \( \alpha = 2/3 \) and is increasing in \( \alpha \). Therefore, we have that
\[
-\frac{2(1-\alpha)}{\tau} + \frac{3\alpha-1}{3-3\alpha} \log (3\alpha-1) - \frac{3\alpha-1}{2} \log (3\alpha-1) \leq -\frac{2(1-\alpha)}{\tau} = L(\delta_\alpha, \alpha).
\]
Therefore, for \( \alpha \in \left( \frac{2}{3}, \frac{3\exp(2/\tau)+1}{3\exp(2/\tau)+3} \right) \), the optimal \( \theta^* \) satisfies \( L(\mu_\theta, \alpha) \leq L(\delta_\alpha, \alpha) \). In particular, solving
\[
\frac{2\sin^2 \theta^*}{\tau} = \log \frac{3\alpha-1}{3-3\alpha}
\]
gives us \( \theta^* = \arcsin \sqrt{\frac{2\log \frac{3\alpha-1}{3-3\alpha}}{\tau}} \).

Therefore, our analysis in these three ranges of \( \alpha \) suggest that the optimal embedding geometry is not collapsed when \( \alpha \geq \frac{2}{3} \). Next, we want to understand when the optimal embedding geometry is not \( \sigma_{d-1} \). A sufficient condition for this is to show that there exists an \( \alpha \geq \frac{2}{3} \) where \( \min_\theta L(\mu_\theta, \alpha) \leq L(\sigma_{d-1}, \alpha) \). We first compute an upper bound on \( \min_\theta L(\mu_\theta, \alpha) \) for \( \alpha > \frac{2}{3} \).

Recall that our loss from (17) can be written as
\[
L(\mu_\theta, \alpha) = -\frac{2(1-\alpha)}{\tau} - 2(1-\alpha) \log \left( \frac{3\alpha-1}{3-3\alpha} \right) = -\frac{2(1-\alpha)}{\tau} - 2(1-\alpha) \log \left( \frac{3\alpha-1}{3-3\alpha} \right).
\] (18)

For ease of notation, let \( x = \frac{3\alpha-1}{3-3\alpha} \in (0, 1) \). We show that \( f(x) = x \log x + (1-x) \log (1-x) \) can be lower bounded quadratically. Performing a Taylor expansion at \( x = 0 \), we have that \( x \log x + (1-x) \log (1-x) \approx -\log 2 + 2(x-1/2)^2 \). We claim that \( x \log x + (1-x) \log (1-x) \geq -\log 2 + 2(x-1/2)^2 \). Note that the two sides are equal when \( x = 1/2 \), so proving this inequality is equivalent to showing that \( f''(x) \geq 4(x-1/2) \) for \( x \geq 1/2 \) and \( f''(x) < 4(x-1/2) \) for \( x < 1/2 \). \( f''(x) = \frac{1}{x^2} \) is equal to \( 4(x-1/2) \) at \( x = 1/2 \), so we want to show that \( f''(x) \geq 4 \) for all \( x \). \( f''(x) = \frac{1}{(1-x)^2} \) satisfies this inequality. Therefore, (18) becomes
\[
L(\mu_\theta, \alpha) \leq -\frac{2(1-\alpha)}{\tau} - 2(1-\alpha) \log \left( \frac{3\alpha-1}{3-3\alpha} \right).
\]

Next, we compute \( L(\sigma_{d-1}, \alpha) \). With \( \sigma_{d-1} \), \( f(x) \) and \( f(x^-) \) are all uniformly distributed on the hypersphere. Therefore,
\[
L(\sigma_{d-1}, \alpha) = \mathbb{E}_x \left[ \log \mathbb{E}_{x^+} \left[ \exp \left( -\|f(x)-f(x^+)^2/2\tau\right) \right] \right] - \frac{1-\alpha}{2\tau} \int \int \|u-u'| \|^2 d\sigma_{d-1}(u) d\sigma_{d-1}(u').
\] (19)

From Wang & Isola (2020), we know that when \( f(x) \) and \( f(x^-) \) are drawn from a distribution with measure \( \sigma_{d-1} \), it holds that \( \mathbb{E}_x \left[ \log \mathbb{E}_{x^+} \left[ \exp \left( -\|f(x)-f(x^+)^2/2\tau\right) \right] \right] = \mathbb{E}_x \left[ \log \mathbb{E}_{x^+} \left[ \exp \left( -\|f(x)-f(x^+)^2/2\tau\right) \right] \right] \).

Define the Gaussian \( \frac{1}{2\tau} \)-energy of \( \sigma_{d-1} \) on \( S^{d-1} \) as \( I_{1/2}[\sigma_{d-1}] = I_{1/2} \int_{S^{d-1}} \int_{S^{d-1}} \exp \left( -\frac{1}{2\tau} \|u-u'|^2 \right) d\sigma_{d-1}(u) d\sigma_{d-1}(u') \).

Then, (19) becomes
\[
L(\sigma_{d-1}, \alpha) = (1-\alpha) \log I_{1/2}[\sigma_{d-1}] + \alpha \log I_{1/2}[\sigma_{d-1}] - \frac{1-\alpha}{2\tau} \int \int \|u-u'|^2 d\sigma_{d-1}(u) d\sigma_{d-1}(u').
\] (20)

From Theorem 4.6.5 of Borodachov et al. (2019), any measure \( \mu \) that has mass centered at the origin, i.e. \( \int u d\mu(u) = 0 \), minimizes the energy \( I_{-2}[\mu] = \int \int \|u-u'|^2 d\mu(u) d\mu(u') \). Therefore, \( \int \int \|u-u'|^2 d\sigma_{d-1}(u) d\sigma_{d-1}(u') \) is equivalent to the energy \( I_{-2}[\sigma_{d-1} + \delta_{\nu_0} + \delta_{\nu_1}] = -\frac{1}{2} \left( \frac{d}{2} \right)^{2} = -2 \), since a measure on two points with probability 1/2 each has mass centered at the origin. Plugging this back into \( L(\sigma_{d-1}, \alpha) \) in (20), we have
\[
L(\sigma_{d-1}, \alpha) = \log I_{1/2}[\sigma_{d-1}] + \frac{1-\alpha}{\tau}.
\] (21)

From Proposition 4.4.1 and Theorem 6.2.1 of Borodachov et al. (2019), \( \sigma_{d-1} \) is the unique equilibrium measure for the Gaussian \( \frac{1}{2\tau} \) kernel on \( S^{d-1} \), and as a result \( I_{1/2}[\sigma_{d-1}] \) is equal to the Wiener constant \( W_{1/2}(S^{d-1}) \). By Proposition A.11.2 of Borodachov et al. (2019), this has the value
\[
W_{1/2}(S^{d-1}) = \frac{2d-2}{\sqrt{\Gamma((d-1)/2)}} \int_{0}^{1} \exp \left( -\frac{2u}{\tau} \right) (u(1-u))^{(d-3)/2} du,
\] (22)

where the Gamma function is \( \Gamma(z) = \int_{0}^{\infty} e^{-z} e^{-x} dx \) for \( z > 0 \).
We bound terms in this probability individually. First, we can write
\[
\xi \sum_{i=1}^{\frac{1}{|P(1, B)|} \sum_{x^+ \in P(1, B)} \log \sigma_f(x, x^+) = 1}
\]
where \(\xi\) is a function of \(\alpha\), and we solve it to get that \(c(\tau, d) \leq 2 + \frac{2}{3} + \frac{2}{3} - 2 \log W_{1/2}^{(S^{d-1})} \).

\[\Box\]

C.2. Proofs for Section 4

**Proposition 2.** Let \(F\) be the set of infinite encoders. Then \(L_{\text{spread}}\) is invariant on class-fixing permutations under \(F\).

**Proof.** We know that any \(f^\pi \in F\) can satisfy \(f^\pi(x_i) = f(x_{\pi(i)})\), since the infinite encoder assumption means that \(f^\pi\) can be arbitrarily fit to any data. Therefore, we only need to show that \(L_{\text{spread}}\) does not change when \(f^\pi\), which permutates within classes, is used instead of \(f\).

For a given batch \(B\) and \(f\), \(L_{\text{spread}}\) is constructed as defined in Section 2.2. The numerator of \(L_{\text{sup}}\) can be written as
\[
\frac{1}{|P(1, B)|} \sum_{i=1}^{\frac{1}{|P(1, B)|} \sum_{x^+ \in P(1, B)} \log \sigma_f(x_i, x^+)}.
\]
This is a summation over the representations of all positive pairs in the batch. Therefore, a permutation \(\pi\) within each class that changes the assignments to the representations will not change the value of this quantity, and
\[
\frac{1}{|P(1, B)|} \sum_{i=1}^{\frac{1}{|P(1, B)|} \sum_{x^+ \in P(1, B)} \log \sigma_f(x_i, x^+)} = \frac{1}{|P(1, B)|} \sum_{i=1}^{\frac{1}{|P(1, B)|} \sum_{x^+ \in P(1, B)} \log \sigma_f(x_i, x^+)}.
\]

Next, the denominator of \(L_{\text{sup}}\) can be written as
\[
\frac{1}{|P(1, B)|} \sum_{i=1}^{\frac{1}{|P(1, B)|} \sum_{x^+ \in P(1, B)} \log \left(\sigma_f(x_i, x^+) + \sum_{x^- \in P(1, B)} \sigma_f(x_i, x^-)\right)}.
\]
Every single positive pair and negative pair is included in this expression, so this quantity is class-fixing permutation invariant.

The numerator of \(L_{\text{cNCE}}\) is \(\frac{1}{|P(1, B)|} \sum_{i=1}^{\frac{1}{|P(1, B)|} \sum_{x^+ \in P(1, B)} \log \left(\sum_{x^+ \in P(1, B)} \sigma_f(x_i, x^+)\right)}\). Since an augmentation of \(x_i\) is a function of \(x_i\) and is disjoint from augmentations of other points, this quantity is class-fixing permutation invariant.

Lastly, the denominator of \(L_{\text{cNCE}}\) is \(\frac{1}{|P(1, B)|} \sum_{i=1}^{\frac{1}{|P(1, B)|} \sum_{x^+ \in P(1, B)} \log \left(\sum_{x^+ \in P(1, B)} \sigma_f(x_i, x^+)\right)}\). From the same logic as the numerator of \(L_{\text{sup}}\), any permutation within the class will still allow each \(x_i\) to be compared with all other points in \(x_i\)’s class, hence being class-fixing permutation invariant.

Therefore, under the infinite encoder assumption where all \(f^\pi\) are valid, \(L_{\text{spread}}\) is permutation invariant.

**Theorem 3.** Denote \(r_f(z, z') = c^2 \delta_f(z, z')^2 - |\text{Var}_f[z] - \text{Var}_f[z']|\). With probability \(1 - \delta\), the coarse-to-fine error is at most
\[
L_{\gamma, f}(z) \leq \frac{\sigma_f(z)}{\sqrt{r_f(z, z') - 2 \log \gamma}} + O\left(\left(\frac{d \log (d/\delta)}{m_z \wedge m_{z'}}\right)^{1/4}\right),
\]
under the boundary condition that \(r_f(z, z') - 2 \log \gamma \geq 16 \sqrt{\frac{2d \log (8d/\delta)}{m_z \wedge m_{z'}} + \frac{2d \log (8d/\delta)}{m_z}}\).

**Proof.** From (6), our loss function can be written as \(L_{\gamma, f}(z) = \text{Pr}_{x \sim P_z}(f(x)^T(W_z - W_{z'})) \leq \log \gamma\). Note that \(|f(x) - W_z|^2 = |f(x)|^2 + |W_z|^2 - 2f(x)^T W_z = 1 + |W_z|^2 - 2f(x)^T W_z\), which means that \(f(x)^T W_z = \frac{1}{2} \left(1 + |W_z|^2 - 2f(x)^T W_z\right)\). We can thus write our loss as
\[
L_{\gamma, f}(z) = \text{Pr}_{x \sim P_z}(f(x)^T(W_z - W_{z'}) \leq \log \gamma)
\]
\[
= \text{Pr}_{x \sim P_z}(\|f(x) - W_z\| \geq \sqrt{\|f(x) - W_z\|^2 - 2 \log \gamma + |W_z|^2 - \|W_{z'}\|^2})^{1/2}).
\]

We bound terms in this probability individually. First, we can write
\[
\|f(x) - W_z\| \leq \|f(x) - E_{x \sim P_z}[f(x)]\| + \xi_z,
\]
where \(\xi_z\) again is constructed as \(\xi_z = \|W_z - E_{x \sim P_z}[f(x)]\| = \left|\frac{1}{m_z} \sum_{x \in D_{z, z}} f_{SC}(x) - E_{x \sim P_z}[f_{SC}(x)]\right|\).
where $\text{Var}_f[z] = \mathbb{E}_{x \sim P_z} \left[ \|f(x) - \mathbb{E}_{x \sim P_{z'}}[f(x)]\|^2 \right]$, and $\text{Var}_f[z']$ is similarly defined. We decompose the first term in (27) and bound it by

$$
\frac{1}{m_z} \sum_{x \in D_{z, s}} \left( \|f(x) - \mathbb{E}_{x \sim P_{s}}[f(x)]\|^2 - \|f(x) - \mathbb{E}_{x \sim P_{s}}[f(x)]\|^2 \right) + \frac{1}{m_{z'}} \sum_{x \in D_{z', s'}} \left( \|f(x) - \mathbb{E}_{x \sim P_{s}}[f(x)]\|^2 - \|f(x) - \mathbb{E}_{x \sim P_{s}}[f(x)]\|^2 \right)
$$

$$
= \frac{1}{m_z} \sum_{x \in D_{z, s}} \left( \|f(x) - \mathbb{E}_{x \sim P_{s}}[f(x)]\|^2 - \|f(x) - \mathbb{E}_{x \sim P_{s}}[f(x)]\|^2 \right) + \zeta_s + \zeta_{s'}
$$

$$
\leq \frac{1}{m_z} \sum_{x \in D_{z, s}} \left( \|f(x) - \mathbb{E}_{x \sim P_{s}}[f(x)]\|^2 - \|f(x) - \mathbb{E}_{x \sim P_{s}}[f(x)]\|^2 \right) + \zeta_s + \zeta_{s'}
$$

$$
\leq \zeta_s + \zeta_{s'}.
$$

where $\zeta_s' = \frac{1}{m_{z'}} \sum_{x \in D_{z', s'}} \left( \|f(x) - \mathbb{E}_{x \sim P_{s'}}[f(x)]\|^2 - \|f(x) - \mathbb{E}_{x \sim P_{s'}}[f(x)]\|^2 \right)$ can be bounded by standard concentration inequalities. Therefore, $\|W_z\|^2 - \|W_{z'}\|^2$ is bounded by

$$
\|W_z\|^2 - \|W_{z'}\|^2 \leq |\text{Var}_f[z] - \text{Var}_f[z']| + 4\zeta_s + 4\zeta_{s'} + \zeta_s + \zeta_{s'}.
$$

(28)
Then, combining (24), (26), and (28), the loss in (23) becomes
\[ L_{\gamma,f}(z) \leq \Pr_{x \sim P_z} \left( \|f(x) - \mathbb{E}_{x \sim P_z}[f(x)]\| \geq (r_f(z,z') - 8\xi_z - 4\xi_z - \zeta_z - 2\log \gamma)^{1/2} - \zeta_z \right). \] (29)

Next, we note that \( \xi_z \leq \sqrt{\frac{2\log(2d/\delta)}{m_z}} \) with probability at least \( 1 - \delta \). Applying Hoeffding’s inequality on \( \xi_z \) gives us \( \zeta_z \leq \sqrt{\frac{8\log(2d/\delta)}{m_z}} \) with probability at least \( 1 - \delta \). Applying a union bound, we have that with probability \( 1 - \delta \), (29) satisfies
\[ L_{\gamma,f}(z) \leq \Pr_{x \sim P_z} \left( \|f(x) - \mathbb{E}_{x \sim P_z}[f(x)]\| \geq r_f(z,z') - 10\sqrt{\frac{2\log(8d/\delta)}{m_z}} - 6\sqrt{\frac{2\log(8d/\delta)}{m_z}} - 2\log \gamma \right)^{1/2} - \sqrt{\frac{2\log(8d/\delta)}{m_z}}. \]

Finally, we can apply Markov’s inequality under the condition that \( r_f(z,z') - 2\log \gamma \geq \frac{2\log(8d/\delta)}{m_z} + 6\sqrt{\frac{2\log(8d/\delta)}{m_z}} + 10\sqrt{\frac{2\log(8d/\delta)}{m_z}}, \) or equivalently \( r_f(z,z') - 2\log \gamma \geq \frac{2\log(8d/\delta)}{m_z} + 16\sqrt{\frac{2\log(8d/\delta)}{m_z \wedge m_z}} \). With probability at least \( 1 - \delta \), our loss is bounded by
\[ L_{\gamma,f}(z) \leq \frac{\mathbb{E}_{x \sim P_z}[f(x) - \mathbb{E}_{x \sim P_z}[f(x)]]}{\sqrt{r_f(z,z') - 2\log \gamma - 16\sqrt{\frac{2\log(8d/\delta)}{m_z \wedge m_z}} - \sqrt{\frac{2\log(8d/\delta)}{m_z}}} \leq \frac{\sigma_f(z)}{\sqrt{r_f(z,z') - 2\log \gamma} + O \left( \frac{d\log(\delta/d)}{m_z \wedge m_z} \right)^{1/4}}. \]

\[ \square \]

**Lemma 1.** Let \( F_{K_L} \) be the class of \( K_L \)-Lipschitz encoders. Then for any \( f_{K_L} \in F_{K_L}, \sigma_{f_{K_L}}(z) \leq K_L \sigma_z. \)

**Proof.** Using Jensen’s inequality and then Lipschitzness of \( f \),
\[ \sigma_f(z) = \mathbb{E}_{x \sim P_z}[\|f(x) - \mathbb{E}_{x \sim P_z}[f(x)]\|] = \mathbb{E}_{x \sim P_z} \left[ \int (f(x) - f(x'))p(x'|z)dx' \right] \]
\[ \leq \mathbb{E}_{x \sim P_z} \left[ \int \|f(x) - f(x')\|p(x'|z)dx' \right] \leq K_L \cdot \mathbb{E}_{x \sim P_z} \left[ \int \|x-x'\|p(x'|z)dx' \right] = K_L \cdot \mathbb{E}_{x \sim P_z} [\|x-x'\|] = K_L \delta_z. \]

\[ \square \]

**Lemma 2.** For any \( g \in G \), suppose there exists a \( K_g > 0 \) such that \( g \) is “reverse Lipschitz”, satisfying \( \|f_{AE}(x) - f_{AE}(x')\| \leq K_g \|g(f_{AE}(x)) - g(f_{AE}(x'))\|, \) and there exists finite \( b \) such that the reconstruction loss satisfies \( \max_x \|g(f_{AE}(x)) - x\|^2 \leq b. \)

Then with probability at least \( 1 - \delta \),
\[ \sigma_{f_{AE}}(z) \leq \frac{2K_g}{p(z|y)} \left( L_{AE}(D_y) + 292^{2_{n_y}}(G \circ F_{AE} \circ id_X) + b \sqrt{\frac{\log(1/\delta)}{2n_y}} \right)^{1/2} + K_g \sigma_z, \]

where \( id_X \) is the identity function on \( X \), and \( p(z|y) = \frac{p(z)}{p(y)} \) is the probability that \( x \) drawn from \( p(\cdot|y) \) has label \( z. \)

**Proof.** We can decompose \( \sigma_{f_{AE}}(z) \) into the following using the assumption on the decoder \( g): \)
\[ \sigma_{f_{AE}}(z) = \mathbb{E}_{x \sim P_z}[\|f_{AE}(x) - f_{AE}(x')\|] \leq \mathbb{E}_{x \sim P_z} \left[ \int \|f_{AE}(x) - f_{AE}(x')\|p(x'|z)dx' \right] \]
\[ \leq \mathbb{E}_{x \sim P_z} \left[ K_g \cdot \|g(f_{AE}(x)) - g(f_{AE}(x'))\|p(x'|z)dx' \right] \]
\[ \leq \mathbb{E}_{x \sim P_z} \left[ K_g (\|g(f_{AE}(x)) - x\| + \|x - x'\| + \|x' - g(f_{AE}(x'))\|)p(x'|z)dx' \right] \]
\[ = 2K_g \mathbb{E}_{x \sim P_z}[\|g(f_{AE}(x)) - x\|] + K_g \mathbb{E}_{x \sim P_z}[\|x - x'\|]. \] (30)
Note that $\mathbb{E}_{x \sim p_y} [\| g(f_{AE}(x)) - x \|] = \sum_{k \in \mathbb{S}_y} \mathbb{E}_{x \sim p_y} [\| g(f_{AE}(x)) - x \|] p(k | y) \geq \mathbb{E}_{x \sim p_y} [\| g(f_{AE}(x)) - x \|] p(z | y)$, so (30) becomes

$$\sigma_{f_{AE}}(z) \leq \frac{2K_g}{p(z | y)} \mathbb{E}_{x \sim p_y} [\| g(f_{AE}(x)) - x \|] + K_g \mathbb{E}_{x \sim p_y} [\| x - x' \|]$$

$$\leq \frac{2K_g}{p(z | y)} \sqrt{\mathbb{E}_{x \sim p_y} [\| g(f_{AE}(x)) - x \|^2] + K_g \mathbb{E}_{x \sim p_y} [\| x - x' \|]}$$

$$= \frac{2K_g}{p(z | y)} \sqrt{\hat{L}_{AE}(n_y) + \mathbb{E}_{x \sim p_y} [\| g(f_{AE}(x)) - x \|^2] - \hat{L}_{AE}(n_y) + K_g \sigma_z}. \quad (31)$$

where $\hat{L}_{AE}(D_y) = \frac{1}{n_y} \sum_{x \in D_y} \| g(f_{AE}(x)) - x \|^2$ is the reconstruction error on the training data $D_y$. We bound the generalization error $\mathbb{E}_{x \sim p_y} [\| g(f_{AE}(x)) - x \|^2] - \hat{L}_{AE}(D_y)$ using Theorem 3.3 of Mohri et al. (2018) to get that with probability at least $1 - \delta$,

$$\sigma_{f_{AE}}(z) \leq \frac{2K_g}{p(z)} \left( \hat{L}_{AE}(D) + 2\mathbb{R}_{n_y}^2(G \circ f_{AE} \circ \text{id}_\mathcal{X}) + b \sqrt{\frac{\log(1/\delta)}{2n_y}} \right)^{1/2} + K_g \sigma_z.$$

Finally, we compare against using a general autoencoder trained on the entire dataset of $n$ points. This yields a bound

$$\sigma_{f_{AE}}(z) \leq \frac{2K_g}{p(z)} \left( \hat{L}_{AE}(D) + 2\mathbb{R}_{n_y}^2(G \circ f_{AE} \circ \text{id}_\mathcal{X}) + b \sqrt{\frac{\log(1/\delta)}{2n}} \right)^{1/2} + K_g \sigma_z,$$

where the only change in the result is that $p(z | y)$ is replaced with $p(z)$, the overall proportion of the subclass, and $n_y$ is replaced with $n$. This highlights a tradeoff: $p(z | y) > p(z)$, but $n_y < n$. A class-conditional autoencoder may suffer from poorer generalization due to lower sample size, but its relative worst case performance on $z$ in expectation is better. On the other hand, a general autoencoder is learned on more data, but its relative worst case performance on $z$ in expectation is worse since the subclass is more rare w.r.t. the training data.

**Lemma 3.** For $a \in \mathcal{A}$ and any $x, x' \in \mathcal{X}$, suppose that $f_{aug} \in F_{aug}$ satisfies $\| f_{aug}(a(x)) - f_{aug}(a(x')) \| \leq K_{aug} \| a(x) - a(x') \|$ for some $K_{aug}$ and that $f(a(x)) = f(x)$ for $x \in \mathcal{D}$. Denote $\sigma^a_{z} = \mathbb{E}_{x,x' \sim p_z} [\| a(x) - a(x') \|]$. Then with probability at least $1 - \delta$,

$$\sigma_{f_{aug}}(z) \leq \frac{2}{p(z)} \left( 2\mathbb{R}_n^2(F_{aug} \circ F_{aug} \circ A) + \sqrt{\frac{2\log(1/\delta)}{n}} \right) + K_{aug} \sigma^a_{z}. \quad (32)$$

**Proof.** We can decompose $\sigma_{f_{aug}}(z)$ into

$$\mathbb{E}_{x \sim p_z} [\| f_{aug}(x) - f_{aug}(x') \|] \leq \mathbb{E}_{x \sim p_z} \left[ \int \| f_{aug}(x) - f_{aug}(x') \| p(x' | z) dx' \right]$$

$$\leq \mathbb{E}_{x \sim p_z} \left[ \int (\| f_{aug}(x) - f_{aug}(a(x)) \| + \| f_{aug}(a(x)) - f_{aug}(a(x')) \| + \| f_{aug}(a(x')) - f_{aug}(x') \|) p(x' | z) dx' \right]$$

$$\leq 2\mathbb{E}_{x \sim p_z} [\| f_{aug}(x) - f_{aug}(a(x)) \|] + K_{aug} \sigma^a_{z}. \quad (32)$$

We can bound $\mathbb{E}_{x \sim p_z} [\| f_{aug}(x) - f_{aug}(a(x)) \|] \leq \frac{1}{p(z)} \mathbb{E} [\| f_{aug}(x) - f_{aug}(a(x)) \|]$. We assume that the encoder is able to satisfy $f_{aug}(x) = f_{aug}(a(x))$ for all training data $x \in \mathcal{D}$, so (32) becomes

$$\sigma_{f_{aug}}(z) \leq \frac{2}{p(z)} \left( \mathbb{E} [\| f_{aug}(x) - f_{aug}(a(x)) \|] - \frac{1}{n} \sum_{i=1}^n \| f(x_i) - f(a(x_i)) \| \right) + K_{aug} \sigma^a_{z}. \quad (33)$$

Then, using Theorem 3.3 from Mohri et al. (2018), with probability at least $1 - \delta$

$$\mathbb{E} [\| f_{aug}(x) - f_{aug}(a(x)) \|] - \frac{1}{n} \sum_{i=1}^n \| f(x_i) - f(a(x_i)) \| \leq 2\mathbb{R}_n^2(F_{aug} \circ F_{aug} \circ A) + \sqrt{\frac{2\log(1/\delta)}{n}}.$$

Therefore, (33) becomes

$$\sigma_{f}(z) \leq \frac{2}{p(z)} \left( 2\mathbb{R}_n^2(F_{aug} \circ F_{aug} \circ A) + \sqrt{\frac{2\log(1/\delta)}{n}} \right) + K_{aug} \sigma^a_{z}. \quad \square$$
D. Additional Theoretical Results

D.1. Optimal $L_{\text{spread}}$ geometry for $K=3$

We provide a proof sketch that when there are $K=3$ classes and $d \geq 3$, there exists a distribution $\mu_{\theta}$ that obtains lower loss than the uniform or collapsed distributions. Synthetic experiments for this setting are in Appendix H (see Figure 7).

For simplicity, let's consider when $d=3$. Without loss of generality, denote the 3-simplex as $v_0 = [1,0,0]$, $v_1 = [-1/2,0,\sqrt{3}/2]$, $v_2 = [-1/2,0,-\sqrt{3}/2]$. We will perform a rotation in the “free” dimension (2), in particular rotating $v_0$ by $\theta$ in the direction orthogonal to the subspace that the simplex is in.

In particular, we construct the rotation matrix $R_{\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Then,

$$R_{\theta}v_0 = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}, \quad R_{\theta}v_1 = \begin{bmatrix} -\cos\theta/2 \\ -\sin\theta/2 \\ \sqrt{3}/2 \end{bmatrix}, \quad R_{\theta}v_2 = \begin{bmatrix} -\cos\theta/2 \\ -\sin\theta/2 \\ -\sqrt{3}/2 \end{bmatrix},$$

and we make a distribution where $\mu_{0,\theta} = \frac{1}{2}\delta_{v_0} + \frac{1}{2}\delta_{R_{\theta}v_0}$, $\mu_{1,\theta} = \frac{1}{2}\delta_{v_1} + \frac{1}{2}\delta_{R_{\theta}v_1}$, $\mu_{2,\theta} = \frac{1}{2}\delta_{v_2} + \frac{1}{2}\delta_{R_{\theta}v_2}$. That is, similar to the binary setting, we take a mixture of a simplex and that simplex rotated by $\theta$ in a dimension orthogonal to its subspace.

Now, we want to compute what the loss is. Recall that our asymptotic loss function is

$$L_{\text{spread}}(\mu, \alpha) = (1-\alpha)\mathbb{E}_x \left[ \log \mathbb{E}_{x^-} \left[ \exp \left( -\frac{1}{2\tau} ||f(x)-f(x^-)||^2 \right) \right] \right] + \alpha \mathbb{E}_x \left[ \log \mathbb{E}_{x^+} \left[ \exp \left( -\frac{1}{2\tau} ||f(x)-f(x^+)||^2 \right) \right] \right] + (1-\alpha)\mathbb{E}_{x,x^+} \left[ \frac{1}{2\tau} ||f(x)-f(x^+)||^2 \right].$$

(34)

For the class collapsed embeddings, note that for $K=3$ the simplex side length is $\sqrt{3}$. Therefore,

$$L_{\text{spread}}(\delta_0, \alpha) = (1-\alpha)\log \exp \left( -\frac{1}{2\tau} \cdot 3 \right) = \frac{-3(1-\alpha)}{2\tau}.$$

Next, we compute the loss for our intermediate distribution. We note the following:

$$\|v_0 - R_{\theta}v_1\|^2 = \|R_{\theta}v_0 - v_1\|^2 = \|v_0 - R_{\theta}v_2\|^2 = \|R_{\theta}v_0 - v_2\|^2 = \left(1 + \frac{\cos\theta}{2}\right)^2 + \frac{\sin^2\theta}{4} + \frac{3}{4}$$

$$\|v_0 - R_{\theta}v_0\|^2 = (1 - \frac{\cos\theta}{2})^2 + \frac{\sin^2\theta}{4} + 3 = \frac{7 - \cos\theta}{2}$$

$$\|v_1 - R_{\theta}v_1\|^2 = \|v_2 - R_{\theta}v_2\|^2 = \left(1 - \frac{\cos\theta}{2}ight)^2 + \frac{\sin^2\theta}{4} = \frac{1 - \cos\theta}{2}$$

and recall that $\|v_1 - v_2\|^2 = \|R_{\theta}v_1 - R_{\theta}v_2\|^2 = 3$. Plugging these back in, we have

$$\mathbb{E}_x \left[ \log \mathbb{E}_{x^-} \left[ \exp \left( -\frac{1}{2\tau} ||f(x)-f(x^-)||^2 \right) / 2\tau \right] \right] = \frac{1}{3} \log \left( \frac{1}{2\tau} \exp \left( -\frac{3}{2\tau} \right) + \frac{1}{2\tau} \exp \left( -\frac{2 + \cos\theta}{2\tau} \right) \right)$$

$$+ \frac{2}{3} \log \left( \frac{1}{2\tau} \exp \left( -\frac{3}{2\tau} \right) + \frac{1}{4\tau} \exp \left( -\frac{7 - \cos\theta}{4\tau} \right) + \frac{1}{4\tau} \exp \left( -\frac{2 + \cos\theta}{2\tau} \right) \right)$$

$$\mathbb{E}_x \left[ \log \mathbb{E}_{x^+} \left[ \exp \left( -\frac{1}{2\tau} ||f(x)-f(x^+)||^2 \right) / 2\tau \right] \right] = \frac{1}{3} \log \left( \frac{1}{2\tau} + \frac{1}{2\tau} \exp \left( -\frac{1 - \cos\theta}{\tau} \right) \right)$$

$$+ \frac{2}{3} \log \left( \frac{1}{2\tau} + \frac{1}{2\tau} \exp \left( -\frac{1 - \cos\theta}{\tau} \right) \right)$$

$$\mathbb{E}_{x,x^+} \left[ \frac{1}{2\tau} ||f(x)-f(x^+)||^2 \right] = \frac{1}{3} \log \left( \frac{1 - \cos\theta}{4\tau} + \frac{1}{6\tau} \right) + \frac{1}{4\tau} = \frac{1 - \cos\theta}{4\tau}$$

We use the above expressions to simplify (34) and numerically check that there exists $\theta$ for $\alpha \gtrsim 0.6$ such that $L(\mu_{\theta}, \alpha) \leq L(\delta_0, \alpha)$. We then numerically check there exists $(\theta, \alpha)$ that also satisfies $L(\mu_{\theta}, \alpha) \leq L(\sigma_{d-1}, \alpha)$, where $L(\sigma_{d-1}, \alpha)$ is defined in (21).
D.2. Permutation Invariance

This result is a simple example of a sufficient condition under which $L_{\text{spread}}$ does not exhibit class-fixing permutation invariance.

**Lemma 4.** Let $\phi : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ be a monotonically increasing function. Suppose that for $x, x' \in \mathcal{X}$, all $f \in \mathcal{F}$ satisfy $\| f(x) - f(x') \| \leq \phi(\| x - x' \|)$. Then, $L_{\text{spread}}$ is not invariant on class-fixing permutations under $\mathcal{F}$.

**Proof.** First, note that $L_{\text{spread}}$ in (2) has terms in the numerator of $L_{\text{map}}$ of the form $\| f(x) - f(x') \|$, where $h(x) = h(x^+)$. We show how to break permutation invariance using this quantity.

Fix two vectors in the embedding space $\mathcal{S}^{d-1}$, $u_a$ and $u_b$. For $x_1, x_2$ in a given class, suppose that $f(x_1) = u_a$ and $f(x_2) = u_b$. We select a third point $x_3$ that is very close to $x_1$, satisfying $\phi(\| x_3 - x_1 \|) < \| u_a - u_b \|$ (this property must hold for some $x_3$ since $\phi$ is monotonic).

We construct a permutation $\pi$ where $\pi(1) = 3, \pi(2) = 1, \pi(3) = 1$. We know that $\| u_a - u_b \| \leq \phi(\| x_1 - x_2 \|)$. Suppose that the mapping $f^\pi$ satisfies $\| f^\pi(x_3) - f^\pi(x_1) \| = \| u_a - u_b \|$. However, this implies that $\| u_a - u_b \| \leq \phi(\| x_3 - x_1 \|)$, which is a contradiction. Therefore, no $f^\pi \in \mathcal{F}$ exists that is able to map the permutation to the same value as $f$ does. As this holds for a single term in $L_{\text{spread}}$, it applies to $L_{\text{spread}}$ overall, demonstrating that such an assumption on $\mathcal{F}$ (which we find is true for a Lipschitz encoder, the autoencoder, and data augmentations) is able to break permutation invariance.

E. Auxiliary Lemmas

**Lemma 5.** Under the infinite encoder assumption, the following statement holds for $L_{\text{diff}}(f)$:

$$\min_{\mu \in \{\mathcal{S}^{d-1}\}} \mathbb{E}_x \left[ \log \mathbb{E}_x - \left[ \exp(-\| f(x) - f(x^-) \|^2/2\sigma) \right] \right] = \min_{\mu \in \{\mathcal{S}^{d-1}\}} \mathbb{E}_x \left[ \log \mathbb{E}_x - \left[ \exp(-\| f(x) - f(x^-) \|^2/2\sigma) \right] \right].$$

**Proof.** Conditioning on the label of $x$ and using the definition of $x^-$ for $K = 2$, we can write $L_{\text{diff}}(f)$ as

$$L_{\text{diff}}(f) = \mathbb{E}_u \left[ \mathbb{E}_x | h(x) = y \left[ \log \mathbb{E}_x - | h(x^-) \neq y \left[ \exp(\sigma f(x, x^-)) \right] \right] \right].$$

$$= \frac{1}{2} \int p(y|y = 0) \left( \log \int p(x^-| y = 1) \exp(\sigma f(x, x^-)) d\mu(x^-) \right) dx$$

$$+ \frac{1}{2} \int p(y|y = 1) \left( \log \int p(x^-| y = 0) \exp(\sigma f(x, x^-)) d\mu(x^-) \right) dx.$$

Since the encoder is assumed to be infinitely powerful, we optimize over the class-conditioned measures $\mu_0$ and $\mu_1$ in $\mathcal{M}(\mathcal{S}^{d-1})$, the set of Borel probability measures on $\mathcal{S}^{d-1}$. The optimization problem is now

$$\min_{\mu_0, \mu_1} \left( \int \log \mathbb{E}_x (\sigma f(x, x^-)) d\mu_1(x) \right) d\mu_0(x) + \left( \int \log \mathbb{E}_x (\sigma f(x, x^-)) d\mu_0(x) \right) d\mu_1(x).$$

Next, define

$$U_{\mu}(u) = \int \exp(u^\top v/\tau) d\mu(v).$$

The expression we want to minimize is thus

$$\min_{\mu_0, \mu_1} \int \log U_{\mu_1}(u) d\mu_0(u) + \int \log U_{\mu_0}(u) d\mu_1(u).$$

Following the approach of Wang & Isola (2020), we analyze the measures $\mu_0, \mu_1$ that minimize this expression in two steps. First, we show that the minimum of (35) exists, i.e. the infimum is attained for some two measures. Second, we show that $U_{\mu_1}$ is constant $\mu_1^\tau$-almost surely, and vice versa. This will allow us to interchange the outer expectation over $x$ and the log in $L_{\text{diff}}(f)$.

1. Minimizers of (35) exist.

Let $m$ be a sequence such that

$$\lim_{m \to \infty} \int \log U_{\mu_1}^m(u) d\mu_0^m(u) + \int \log U_{\mu_0}^m(u) d\mu_1^m(u) = \inf_{\mu_0, \mu_1} \int \log U_{\mu_1}(u) d\mu_0(u) + \int \log U_{\mu_0}(u) d\mu_1(u).$$
Using Helly’s Selection Theorem twice, there exists a subsequence $n$ such that $\{ (\mu_0^n, \mu_1^n) \}_{n}$ converges to a weak cluster point $(\mu_0^{*}, \mu_1^{*})$. Because $\{ \log U_{\mu_0^n} \}_{n}$ is uniformly bounded and continuously convergent to $\log U_{\mu_0^{*}}$ and same for $\mu_1^n$ and $\mu_1^{*}$, it holds that

$$\int \log U_{\mu_1^n}(u) d\mu_0^n(u) + \int \log U_{\mu_1^n}(u) d\mu_1^n(u)$$

$$= \lim_{n \to \infty} \int \log U_{\mu_1^n}(u) d\mu_0^n(u) + \int \log U_{\mu_1^n}(u) d\mu_1^n(u).$$

and therefore $\mu_0^{*}, \mu_1^{*}$ achieve the infimum of (35).

2. $U_{\mu_1^n}$ is constant $\mu_0^n$-almost surely and $U_{\mu_0^n}$ is constant $\mu_1^n$-almost surely, for any minimizer $(\mu_0^n, \mu_1^n)$ of (35).

Formally, define $(\mu_0^{*}, \mu_1^{*})$ to be a solution of (35), i.e.

$$\mu_0^{*}, \mu_1^{*} \in \arg \min_{\mu_0, \mu_1} \int \log U_{\mu_1}(u) d\mu_0(u) + \int \log U_{\mu_0}(u) d\mu_1(u).$$

Define the Borel sets where $\mu_i^{*}$ has positive measure to be $\mathcal{T}_i = \{ T \in \mathcal{M}(\mathbb{S}^{d-1}) : \mu_i^{*}(T) > 0 \}$. Define the conditional distribution of $\mu_i^{*}$ on $T$ for some $T \in \mathcal{T}_i$ as $\mu_{i,T}^{*}(A) = \frac{\mu_i^{*}(A \cap T)}{\mu_i^{*}(T)}$.

Now we consider a mixture $(1-\alpha)\mu_0 + \alpha\mu_0^{*,-T}$. The first variation of $\mu_0^{*}$ states that

$$0 = \frac{\partial}{\partial \alpha} \left[ \int \log U_{\mu_1^{*}}(u) d((1-\alpha)\mu_0 + \alpha\mu_0^{*,-T})(u) + \int \log U_{((1-\alpha)\mu_0 + \alpha\mu_0^{*,-T})^{*}}(u) \right]_{\alpha=0}$$

$$= \int \log U_{\mu_1^{*}}(u) d(\mu_0^{*} - \mu_0^{*})(u) + \int \frac{U_{\mu_0^{*,-T}}(u) - U_{\mu_0^{*}}(u)}{U_{\mu_0^{*}}(u)} d\mu_1^{*}(u),$$

Where we’ve used the fact that $\frac{\partial}{\partial \alpha} \left[ \int \exp(u^Tv/\tau) d((1-\alpha)\mu_0 + \alpha\mu_0^{*,-T})(v) \right]_{\alpha=0} = U_{\mu_0^{*,-T}}(u) - U_{\mu_0^{*}}(u)$. Therefore, due to symmetry the optimality conditions using the first variation are

$$\int \log U_{\mu_1^{*}}(u) d(\mu_0^{*} - \mu_0^{*})(u) + \int \frac{U_{\mu_0^{*,-T}}(u) - U_{\mu_0^{*}}(u)}{U_{\mu_0^{*}}(u)} d\mu_1^{*}(u) = 1$$

(36)

$$\int \log U_{\mu_1^{*}}(u) d(\mu_1^{*} - \mu_1^{*})(u) + \int \frac{U_{\mu_0^{*,-T}}(u) - U_{\mu_0^{*}}(u)}{U_{\mu_1^{*}}(u)} d\mu_0^{*}(u) = 1$$

(37)

Now, let $\{ T_0^n \}_{n=1}^{\infty}$ be a sequence of sets in $\mathcal{T}_0$ such that

$$\lim_{n \to \infty} \int U_{\mu_1^{*}}(u) d\mu_0^{*,-T_0^n}(u) = \sup_{T_0 \in \mathcal{T}_0} \int U_{\mu_1^{*}}(u) d\mu_0^{*,-T_0}(u) = U_{1,0}^{*}.$$ 

and similarly let $\{ T_1^n \}_{n=1}^{\infty}$ be a sequence of sets in $\mathcal{T}_1$ such that

$$\lim_{n \to \infty} \int U_{\mu_0^{*}}(u) d\mu_1^{*,-T_1^n}(u) = \sup_{T_1 \in \mathcal{T}_1} \int U_{\mu_0^{*}}(u) d\mu_1^{*,-T_1}(u) = U_{0,1}^{*}.$$ 

It holds that $\mu_0^{*} \{ u : U_{\mu_1^{*}}(u) \geq U_{1,0}^{*} \} = 0$, $\mu_0^{*,-T_0^n} \{ u : U_{\mu_1^{*}}(u) \geq U_{1,0}^{*} \} = 0$ and similarly $\mu_1^{*} \{ u : U_{\mu_0^{*}}(u) \geq U_{0,1}^{*} \} = 0$, $\mu_1^{*,-T_1^n} \{ u : U_{\mu_0^{*}}(u) \geq U_{0,1}^{*} \} = 0$.

This implies that asymptotically $U_{\mu_0^{*}}$ is constant $\mu_1^{*,-T_1^n}$-almost surely:

$$\int \left| U_{\mu_0^{*}}(u) - U_{\mu_0^{*}}(u') \right| d\mu_1^{*,-T_1^n}(u)$$

$$\leq 2 \max \left( 0, U_{\mu_0^{*}}(u) - U_{\mu_0^{*}}(u') \right) d\mu_1^{*,-T_1^n}(u)$$

$$\leq 2 \left( U_{0,1}^{*} - U_{\mu_0^{*}}(u) \right) d\mu_1^{*,-T_1^n}(u) \to 0.$$ 

And the same holds that $U_{\mu_1^{*}}$ is constant $\mu_0^{*,-T_0^n}$-almost surely. As a result, $\lim_{n \to \infty} \int \log U_{\mu_0^{*}}(u) d\mu_1^{*,-T_1^n}(u) = \log U_{0,1}$ and $\lim_{n \to \infty} \int \log U_{\mu_1^{*}}(u) d\mu_0^{*,-T_0^n}(u) = \log U_{1,0}^{*}$. 

We now revisit (36) with a mixture over $\mu_0^*$ and $\mu_{0,T^2}$:

$$1 = \int \log U_{\mu_1}(u) d(\mu_{0,T^2}^* - \mu_0^*)(u) + \int \frac{U_{\mu_1}(u)}{U_{\mu_0}(u)} d\mu_0^*(u) 
\geq \int \log U_{\mu_1}(u) d(\mu_{0,T^2}^* - \mu_0^*)(u) + \frac{1}{U_{0,1}} \int U_{\mu_1}(u) d\mu_{0,T^2}^*(u).$$

Taking the limit of both sides as $n \to \infty$, we get

$$1 \geq \log U_{1,0}^* - \int \log U_{\mu_1}(u) d\mu_0^*(u) + \frac{1}{U_{0,1}} U_{1,0}^*.$$ 

and rearranging and doing the same to (37) yields

$$1 - \frac{U_{1,0}^*}{U_{0,1}^*} \geq \log U_{1,0}^* - \int \log U_{\mu_1}(u) d\mu_0^*(u)$$

Note that Jensen’s inequality and the definition of $U_{1,0}^*$ tell us that $\int \log U_{\mu_1}(u) d\mu_0^*(u) \leq \log \int U_{\mu_1}(u) d\mu_0^*(u) \leq \log U_{1,0}^*$, which means that $1 - \frac{U_{1,0}^*}{U_{0,1}^*} \geq 0$. However, applying the same logic also tells us that $1 - \frac{U_{1,0}^*}{U_{0,1}^*} \geq 0$. The only case in which this is possible is when $U_{1,0}^* = U_{0,1}^*$, in which case equality is obtained. Therefore, this means that for optimal $\mu_0^*, \mu_1^*$, it holds that

$$\int \log U_{\mu_1}(u) d\mu_0^*(u) = \int U_{\mu_1}(u) d\mu_0^*(u) \quad (38)$$

$$\int \log U_{\mu_0}(u) d\mu_1^*(u) = \int U_{\mu_0}(u) d\mu_1^*(u) \quad (39)$$

Using (38) and (39), minimizing (35) is equivalent to minimizing

$$\log \int U_{\mu_1}(u) d\mu_0(u),$$

where we use the fact that $\mu_0$ and $\mu_1$ are interchangeable in the above expression. This expression can be written as $\log \mathbb{E}_{x,x'} \left[ \exp(-\|f(x) - f(x')\|^2/2\tau) \right]$, which completes our proof.

\[ \Box \]

**Lemma 6.** Suppose $\mathcal{F}$ is a family of functions mapping from $\mathcal{X}$ to $\mathbb{R}^d$. Define $f(x)[j]$ as the $j$th element of $f(x)$ and suppose that for all $j$, $|f(x)[j]| \leq b$. Define the element-wise class $\mathcal{F}_j = \{ f(\cdot)[j] : f \in \mathcal{F} \}$. Then, with probability at least $1 - \delta$ over $n$ i.i.d. samples $\{x_i\}_{i=1}^n$,

$$\left\| \mathbb{E}[f(x)] - \frac{1}{n} \sum_{i=1}^n f(x_i) \right\| \leq 2 \mathfrak{R}_n(\mathcal{F}) + bd \sqrt{\frac{\log(d/\delta)}{2n}} \quad \forall f \in \mathcal{F},$$

where $\mathfrak{R}_n(\mathcal{F}) = \sum_{j=1}^d \mathfrak{R}_n(\mathcal{F}_j)$.

**Proof.** Using the triangle inequality,

$$\left\| \mathbb{E}[f(x)] - \frac{1}{n} \sum_{i=1}^n f(x_i) \right\| = \left( \sum_{j=1}^d \left( \mathbb{E}[f(x)[j]] - \frac{1}{n} \sum_{i=1}^n f(x_i)[j] \right)^2 \right)^{1/2} \leq \sum_{j=1}^d \left\| \mathbb{E}[f(x)[j]] - \frac{1}{n} \sum_{i=1}^n f(x_i)[j] \right\|.$$

Using Theorem 3.3 of Mohri et al. (2018), we know that with probability at least $1 - \delta$,

$$\mathbb{E}[f(x)[j]] \leq \frac{1}{n} \sum_{i=1}^n f(x_i)[j] + 2 \mathfrak{R}_n(\mathcal{F}_j) + b \sqrt{\frac{2 \log(1/\delta)}{n}}.$$

Applying a union bound, we have that with probability at least $1 - \delta$,

$$\left\| \mathbb{E}[f(x)] - \frac{1}{n} \sum_{i=1}^n f(x_i) \right\| \leq \sum_{j=1}^d \left( 2 \mathfrak{R}_n(\mathcal{F}_j) + b \sqrt{\frac{\log(d/\delta)}{2n}} \right) = 2 \mathfrak{R}_n(\mathcal{F}) + bd \sqrt{\frac{2 \log(d/\delta)}{n}}.$$ 

\[ \Box \]
F. Additional Experimental Details

We describe details about the datasets, model architectures, and hyperparameters.

F.1. Datasets

We first describe all the datasets in more detail:

- **CIFAR10, CIFAR100, and MNIST** are all the standard computer vision datasets.

- **CIFAR10-Coarse** consists of two superclasses: animals (dog, cat, deer, horse, frog, bird) and vehicles (car, truck, plane, boat).

- **CIFAR100-Coarse** consists of twenty superclasses. We artificially imbalance subclasses to create **CIFAR100-Coarse-U**. For each superclass, we select one subclass to keep all 500 points, select one subclass to subsample to 250 points, select one subclass to subsample to 100 points, and select the remaining two to subsample to 50 points. We use the original CIFAR100 class index to select which subclasses to subsample: the subclass with the lowest original class index keeps all 500 points, the next subclass keeps 250 points, etc.

- **TinyImageNet-Coarse** (Le & Yang, 2015) consists of 67 superclasses constructed from the ImageNet class hierarchy (Bostock, 2018). The 67 superclasses are as follows: arachnid, armadillo, bear, bird, bug, butterfly, cat, coral, crocodile, crustacean, dinosaur, dog, echinoderms, ferret, fish, flower, frog, fruit, fungus, hog, lizard, marine mammals, marsupial, mollusk, mongoose, monotreme, person, plant, primate, rabbit, rodent, salamander, shark, sloth, snake, trilobite, turtle, ungulate, vegetable, wild cat, wild dog, accessory, aircraft, ball, boat, building, clothing, container, cooking, decor, electronics, fence, food, furniture, hat, instrument, lab equipment, other, outdoor scene, paper, sports equipment, technology, tool, toy, train, vehicle and weapon.

- **MNIST-Coarse** consists of two superclasses: <5 and ≥5.

- **Waterbirds** (Sagawa et al., 2019) is a robustness dataset designed to evaluate the effects of spurious correlations on model performance. The waterbirds dataset is constructed by cropping out birds from photos in the Caltech-UCSD Birds dataset (Welinder et al., 2010a), and pasting them on backgrounds from the Places dataset (Zhou et al., 2014). It consists of two categories: water birds and land birds. The water birds are heavily correlated with water backgrounds and the land birds with land backgrounds, but 5% of the water birds are on land backgrounds, and 5% of the land birds are on water backgrounds. These form the (imbalanced) hidden strata.

- **ISIC** is a public skin cancer dataset for classifying skin lesions (Codella et al., 2019) as malignant or benign. 48% of the benign images contain a colored patch, which form the hidden strata.

- **CelebA** is an image dataset commonly used as a robustness benchmark (Liu et al., 2015; Sagawa et al., 2019). The task is blonde/not blonde classification. Only 6% of blonde faces are male, which creates a rare stratum in the blonde class.

F.2. Model Architectures

We use a ViT model (Dosovitskiy et al., 2020) (4 x 4 patch size, 7 multi-head attention layers with 8 attention heads and hidden MLP size of 256, final embedding size of 128) as the encoder for the transfer learning experiments and a ResNet50 for the robustness experiments. For the ViT models, we jointly optimize the contrastive loss with a cross-entropy loss head. For the ResNets, we train the contrastive loss on its own and use linear probing on the final layer.

For the autoencoder, we use the same encoder backbone as the main model, and use a ResNet18 in reverse order for the decoder. The convolutions are replaced with resize convolutions. We use the implementation in PyTorch Lightning Bolts\(^2\) (Falcon & Cho, 2020).

\(^2\)https://github.com/PyTorchLightning/lightning-bolts/blob/master/pl_bolts/models/autoencoders/components.py
Perfectly Balanced: Improving Transfer and Robustness of Supervised Contrastive Learning

<table>
<thead>
<tr>
<th>Dataset</th>
<th>InfoNCE</th>
<th>$L_{sup}$</th>
<th>$L_{spread}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>89.7</td>
<td>90.9</td>
<td>91.5</td>
</tr>
<tr>
<td>CIFAR10-Coarse</td>
<td>97.7</td>
<td>96.5</td>
<td>98.1</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>68.0</td>
<td>67.5</td>
<td>69.1</td>
</tr>
<tr>
<td>CIFAR100-Coarse</td>
<td>76.9</td>
<td>77.2</td>
<td>78.3</td>
</tr>
<tr>
<td>CIFAR100-Coarse-U</td>
<td>72.1</td>
<td>71.6</td>
<td>72.4</td>
</tr>
<tr>
<td>MNIST</td>
<td>99.1</td>
<td>99.2</td>
<td></td>
</tr>
<tr>
<td>MNIST-Coarse</td>
<td>99.1</td>
<td>99.4</td>
<td>99.4</td>
</tr>
<tr>
<td>Waterbirds</td>
<td>77.8</td>
<td>73.9</td>
<td>77.9</td>
</tr>
<tr>
<td>ISIC</td>
<td>87.8</td>
<td>88.7</td>
<td>90.0</td>
</tr>
</tbody>
</table>

Table 6. End model performance training with $L_{spread}$ on various datasets compared against contrastive baselines. All metrics are accuracy except for ISIC (AUROC). $L_{spread}$ produces the best performance in 7 out of 9 cases, and matches the best performance in 1 case.

<table>
<thead>
<tr>
<th>Method</th>
<th>CIFAR10</th>
<th>CIFAR100</th>
<th>CIFAR100-U</th>
<th>MNIST</th>
<th>TinyImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baselines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross Entropy</td>
<td>71.1 ± 0.2</td>
<td>54.2 ± 0.2</td>
<td>56.4 ± 0.4</td>
<td>98.7 ± 0.1</td>
<td>44.4 ± 0.1</td>
</tr>
<tr>
<td>InfoNCE (Chen et al., 2020a)</td>
<td>77.6 ± 0.1</td>
<td>60.5 ± 0.1</td>
<td>56.4 ± 0.3</td>
<td>98.4 ± 0.1</td>
<td>44.9 ± 0.1</td>
</tr>
<tr>
<td>SupCon (Khosla et al., 2020)</td>
<td>51.8 ± 1.2</td>
<td>56.1 ± 0.1</td>
<td>49.8 ± 0.3</td>
<td>95.4 ± 0.1</td>
<td>43.9 ± 0.1</td>
</tr>
<tr>
<td>SupCon + InfoNCE (Islam et al., 2021)</td>
<td>77.6 ± 0.1</td>
<td>55.7 ± 0.1</td>
<td>48.0 ± 0.2</td>
<td>98.6 ± 0.1</td>
<td>46.1 ± 0.1</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cAuto</td>
<td>71.4 ± 0.1</td>
<td>62.9 ± 0.1</td>
<td>58.7 ± 0.5</td>
<td>98.7 ± 0.1</td>
<td>47.1 ± 0.1</td>
</tr>
<tr>
<td>SupCon + cNCE ($L_{spread}$)</td>
<td>77.1 ± 0.1</td>
<td>58.7 ± 0.2</td>
<td>53.5 ± 0.4</td>
<td>98.5 ± 0.1</td>
<td>45.8 ± 0.1</td>
</tr>
<tr>
<td>SupCon + cAuto</td>
<td>71.7 ± 0.1</td>
<td>63.8 ± 0.6</td>
<td><strong>59.8 ± 0.3</strong></td>
<td>98.7 ± 0.1</td>
<td>49.3 ± 0.1</td>
</tr>
<tr>
<td>SupCon + cNCE + cAuto (THANOS)</td>
<td><strong>79.1 ± 0.2</strong></td>
<td><strong>65.0 ± 0.2</strong></td>
<td>59.7 ± 0.3</td>
<td><strong>99.0 ± 0.1</strong></td>
<td><strong>49.6 ± 0.1</strong></td>
</tr>
</tbody>
</table>

Table 7. Coarse-to-fine transfer learning performance (expanded table). Best in bold.

F.3. Hyperparameters

For the coarse dataset training, all models were trained for 600 epochs with an initial learning rate of 0.0003, a cosine annealing learning rate scheduler with $T_{max}$ set to 100 and the AdamW optimizer. A dropout rate of 0.05 was used. We did not use weight decay. For each coarse dataset, we trained 5 separate models which jointly optimize a cross-entropy loss head with either a contrastive loss (InfoNCE, SupCon, SupCon + InfoNCE, SupCon + Class-conditional InfoNCE) or a reconstruction loss (mean squared error).

In the coarse-to-fine transfer experiments, we trained 5 separate models for each of the configurations reported in Table 3 using 5 random seeds (42, 32, 64, 128 and 72). All models were trained for 100 epochs with an initial learning rate of 0.001, a cosine annealing learning rate scheduler with $T_{max}$ set to 100 and the AdamW optimizer. All transfer experiments were run using Tesla V100 machines.

All experiments were run using a batch size of 128 for both training and evaluation.

G. Additional Experimental Results

We present additional experimental results on end model accuracy, transfer with cross entropy, more datasets, more baselines, and full ablations.

G.1. End Model Accuracy

See Table 6 for raw accuracy. We confirm that using $L_{spread}$ instead of $L_{sup}$ does not degrade end model performance.

G.2. Additional Transfer Results

We reproduce Table 3 and additional report the performance of training with cross entropy loss (Table 7).
Table 8. Coarse-to-fine transfer learning performance on two additional datasets. Best in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Caltech-UCSD Birds</th>
<th>Stanford Dogs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross Entropy</td>
<td>8.2</td>
<td>14.9</td>
</tr>
<tr>
<td>SupCon (Khosla et al., 2020)</td>
<td>7.8</td>
<td>15.0</td>
</tr>
<tr>
<td>cAuto</td>
<td>8.8</td>
<td>17.7</td>
</tr>
<tr>
<td>SupCon + cNCE ($L_{spread}$)</td>
<td>7.5</td>
<td>16.5</td>
</tr>
<tr>
<td>SupCon + cAuto</td>
<td>9.1</td>
<td>19.8</td>
</tr>
<tr>
<td>SupCon + cNCE + cAuto (THANOS)</td>
<td>8.8</td>
<td>20.8</td>
</tr>
</tbody>
</table>

Table 9. Coarse-to-fine transfer learning performance on CIFAR10 with two additional baselines. Best in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>CIFAR10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clip Positives</td>
<td>35.3</td>
</tr>
<tr>
<td>Weighted Pos in Denom.</td>
<td>56.3</td>
</tr>
<tr>
<td>THANOS</td>
<td>79.1</td>
</tr>
</tbody>
</table>

Table 10. Ablations on the autoencoder and data augmentation on CIFAR10 coarse-to-fine transfer.

<table>
<thead>
<tr>
<th>General vs. Class-Conditional Autoencoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>gAuto</td>
</tr>
<tr>
<td>cAuto</td>
</tr>
<tr>
<td>SupCon</td>
</tr>
<tr>
<td>SupCon + gAuto</td>
</tr>
<tr>
<td>SupCon + cAuto</td>
</tr>
<tr>
<td>SupCon + cNCE</td>
</tr>
<tr>
<td>SupCon + cNCE + gAuto</td>
</tr>
<tr>
<td>SupCon + cNCE + cAuto (THANOS)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cNCE With and Without Augmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SupCon + cNCE - augmentation</td>
</tr>
<tr>
<td>SupCon + cNCE</td>
</tr>
</tbody>
</table>

G.3. Additional Datasets

Table 8 report the performance of C2F transfer on two additional datasets, Caltech-UCSD Birds (Welinder et al., 2010b) and Stanford Dogs (Khosla et al., 2011).

G.4. Additional Baselines

Table 9 reports C2F transfer performance with two additional baselines—a) clipping the values of the positives in the numerator $L_{sup}$, and b) upweighting the negatives in the denominator of $L_{sup}$. Both these methods underperform THANOS.

G.5. Ablations and Sensitivity Studies

In this section, we validate our specific theoretical claims on the class-conditional autoencoder, data augmentation, and the Lipschitzness of the decoder.

We use two ablations to validate our claims that the class-conditional autoencoder outperforms a generic autoencoder, and that data augmentation in the class-conditional InfoNCE loss is critical for inducing subclass clustering. Table 10 reports the results:

- Lemma 2 claims that a class-conditional autoencoder should outperform a generic autoencoder in coarse-to-fine transfer. Indeed, we find that using a generic autoencoder underperforms a class-conditional autoencoder by 30.0 points on CIFAR10 coarse-to-fine transfer. Furthermore, the generic autoencoder does not improve performance of SupCon or its variants as well; we observe average lift of 2.0 points, compared to 11.0 points for the class-conditional autoencoder.
Figure 3. Measures of Lipschitzness for three ways to break permutation invariance. Encoder Lipschitzness reports pixel distance on the X axis and embedding distance on the Y axis. The Decoder reports pixel distance of the reconstruction. For augmentations, we run ten augmentations and pick the pair with the smallest ratio of pixel distance to embedding distance. The decoder is more Lipschitz than the encoder, and the encoder is more Lipschitz under augmentations than under traditional Lipschitzness.

- Lemma 3 claims that data augmentation in the class-conditional InfoNCE loss is key to break the permutation invariance. Removing data augmentation degrades performance by 35.4 points (and produces the permutation shown in Figure 1).

Finally, Figure 3 measures the Lipschitzness of an encoder trained with $L_{spread}$ and the reverse Lipschitzness of the decoder from a class-conditional autoencoder. The encoder displays a high Lipschitzness constant (not very Lipschitz). However, it displays a low Lipschitzness constant over augmentations. The decoder displays a lower reverse Lipschitzness constant. This suggests that the assumptions in Lemmas 2 and 3 are reasonable.

To measure Lipschitzness of an encoder, we measure distance in embedding space of an encoder trained with $L_{spread}$ vs. distance in pixel space of two images (blue line). To measure reverse Lipschitzness of a decoder, we make the same measurement, but over pixel distance of decoded images from an autoencoder (orange line). To measure Lipschitzness under data augmentations, we measure the minimum ratio between embedding distance and pixel distance for 10 randomly-generated augmentations of two images (red line). The Lipschitzness constants $K_L$, $K_g$, and $K_{aug}$ in Table 1 are the slopes of the lines tangent to each of the curves in Figure 3 from the origin.

H. Synthetic Experiments

We conduct synthetic experiments to understand the optimal geometry that minimizes the asymptotic loss $L_{spread}(\mu, \alpha)$ as defined in Section 3.2.

**Setup** We minimize an empirical estimate of the asymptotic loss over a set of unit vectors $\{u_i\}_{i=1}^{Kn_y}$. Denote $u$ as a unit vector, and denote $h(u)$ as its class label. The loss we minimize is

$$
(1 - \alpha) \frac{1}{Kn_y} \sum_{i=1}^{Kn_y} \log \left( \frac{1}{(K-1)n_y: h(u_i) \neq h(u_j)} \sum_{j: h(u_j) = h(u_i)} \exp(-\|u_i - u_j\|^2/2\tau) \right)
+ \alpha \cdot \frac{1}{Kn_y} \sum_{i=1}^{Kn_y} \log \left( \frac{1}{n_y: h(u) = h(u_i)} \sum_{j: h(u_j) = h(u_i)} \exp(-\|u_i - u_j\|^2/2\tau) \right) + (1 - \alpha) \frac{1}{Kn_y^2} \sum_{h(u) = h(u')} \|u - u'\|^2/2\tau.
$$

(40)

We use scipy.minimize and the Sequential Least Squares Programming (SLSQP) option. We report the set of vectors that obtain the lowest loss over 5 runs with random initializations (seeds $0-4$) as the optimal geometry.

We compute an empirical estimate of $sf(y)$ as $\frac{1}{n_y} \sum_{u: h(u) = y} \left\| u - \frac{1}{n_y} \sum_{u': h(u') = y} u' \right\|$, and average over all classes.

**Matching $\mu$ and the optimal geometry** Figure 4 displays our constructed distribution $\mu_\theta$ as well as simulations on $S^1$ for $K = 2$. In particular, the right figure consists of the optimal geometry for $n_y = 20$, $\tau = 0.5$. We see that for $\alpha = 0.6$ (which is below our threshold in Theorem 2), that the optimal geometry is collapsed. For $\alpha = 0.7$, the optimal geometry appears
Perfectly Balanced: Improving Transfer and Robustness of Supervised Contrastive Learning

Figure 4. Left: Distribution family $\mu_\theta$. Right: Simulations of optimal geometry for binary setting on $S^1$.

to closely match the parametrization of $\mu_\theta$. For $\alpha = 0.8$, which is above the theoretical threshold, the optimal geometry is uniform per class.

Figure 5. The value of $c(\tau,d)$, which determines the range of $\alpha$ for which $\mu_\theta$ obtains lower loss than $\delta_\theta$ and $\sigma_{d-1}$.

Computing $c(\tau,d)$ Next, we compute the value of $c(\tau,d)$, the constant in Theorem 2, over values of $\tau$ and $d$ to verify that $\alpha \in (2/3,c(\tau,d))$ is a valid range for which the optimal geometry is neither collapsed nor uniform. This quantity depends on the Wiener constant of the Gaussian $\frac{1}{2\tau}$-energy on $S^{d-1}$, which does not have a closed form expression (see (22)). Figure 5 shows that $c(\tau,d) > 2/3$ for $d$ up to 128 (which is the dimension of our embedding space) and for $\tau = 0.1,0.25,0.5,1,2$.

Varying $\tau$ and $d$ for $K = 2$ We plot $\alpha$ versus spread $s_f(y)$ for the optimal geometry in the binary setting, and in particular we vary $\tau$ and $d$ in Figure 6. We compute the optimal geometries over $\alpha = 0.5,0.6,0.67,0.69,0.71,0.73,0.75,0.8,0.9$. Figure 6 left shows how the spread changes as $\alpha$ increases for dimensions $d = 2,5,10$ and $n_y = 8$ samples with $\tau = 0.25$, and the right shows how the spread changes as $\alpha$ increases for $\tau = 0.1,0.25,0.5,1,2$ with $d = 2$. Note that for all dimensions and all $\tau$, the optimal geometry has nonzero spread starting at $\alpha = 0.67$, matching our theoretical findings. The point at which the uniform distribution
becomes optimal is less clear but follows the trends we note from Figure 5 for $c(\tau, d)$. This figure matches our findings that an $\alpha$ that induces appropriate spread exists over a certain range, outside of which behavior is strictly collapsed or uniform.

**Figure 7.** Visualizations of the optimal geometry for $n_y = 8, K = 3$ in $S^2$ across various $\alpha$.

**Multiclass Analysis for $K = 3$** Similar to Figure 4, we show for $K = 3$ and $S^2$ that the optimal geometry is collapsed for low $\alpha$, sufficiently spread for a particular range, and uniform for high $\alpha$. Figure 7 displays the optimal geometry for $n_y = 8, \tau = 0.5$ across $\alpha = 0.6, 0.64, 0.8$, suggesting that the multiclass case exhibits similar behavior as $\alpha$ varies.

**Figure 8.** The spread $s_f(y)$ of the optimal geometry for a given $\alpha$ when $K = 3, n_y = 8$. Left: how optimal spread changes based on dimension $d$ of the embedding space. Right: how optimal spread changes based on the temperature hyperparameter $\tau$.

We plot $\alpha$ versus spread $s_f(y)$ for the optimal geometry in the multiclass setting, and again we vary $\tau$ and $d$ in Figure 8. We see that the behavior of the optimal geometry’s spread $s_f(y)$ across $\alpha$ is roughly similar to that of $K = 2$ in Figure 6.