THE DETAILS MATTER: PREVENTING CLASS COLLAPSE IN SUPERVISED CONTRASTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised contrastive learning optimizes a loss that pushes together embeddings of points from the same class while pulling apart embeddings of points from different classes. Class collapse—when every point from the same class has the same embedding—minimizes this loss but loses critical information that is not encoded in the class labels. For instance, the "cat" label does not capture unlabeled categories such as breeds, poses, or backgrounds (which we call "strata"). As a result, class collapse produces embeddings that are less useful for downstream applications such as transfer learning and achieves sub-optimal generalization error when there are strata. We explore a simple modification to supervised contrastive loss that prevents class collapse by uniformly pulling apart individual points from the same class. More importantly, we introduce a theoretical framing to analyze this loss through a view of how it embeds strata of different sizes. We show that our loss maintains distinctions between strata in embedding space, even though it does not explicitly use strata labels. We empirically explore several downstream implications of this insight. Our loss produces embeddings that achieve lift on three downstream applications by distinguishing strata: 4.4 points on coarse-to-fine transfer learning, 2.5 points on worst-group robustness, and 1.0 points on minimal coreset construction. Our loss also produces more accurate models, with up to 4.0 points of lift across 9 tasks.

1 INTRODUCTION

Supervised contrastive learning has emerged as a promising method for training deep models, with strong empirical results over traditional supervised learning (Khosla et al., 2020). Recent theoretical work has shown that under certain assumptions, *class collapse*—when the representation of every point from a class collapses to the same embedding on the hypersphere, as in Figure 1—minimizes the supervised contrastive loss L_{SC} (Graf et al., 2021). And modern deep networks, which can memorize arbitrary labels (Zhang et al., 2016), are powerful enough to produce class collapse.

Although class collapse minimizes L_{SC} and produces accurate models, it loses information that is not explicitly encoded in the class labels. For example, consider images with the label "cat." As shown in Figure 1, some cats may be sleeping, some may be jumping, and some may be swatting at a bug. We call each of these semantically-unique categories of data—some of which are rarer than others, and none of which are explicitly labeled—a *stratum*. Distinguishing strata is important; it empirically can improve model performance (Hoffmann et al., 2001) and fine-grained robustness (Sohoni et al., 2020), and it is also critical in applications such as medical imaging (Oakden-Rayner et al., 2020). But L_{SC} maps the sleeping, jumping, and swatting cats all to a single "cat" embedding, losing strata information. As a result, these embeddings are less useful for common downstream applications in the modern machine learning landscape, such as transfer learning.

In this paper, we explore a simple modification to L_{SC} that prevents class collapse. We introduce a theoretical framing to understand how this modification affects embedding quality by studying how it embeds strata in embedding space. We evaluate our loss both in terms of embedding quality, which we evaluate through three downstream applications, and end model quality.

In Section 3, we present our modification to L_{SC}, which prevents class collapse by changing how embeddings are pushed and pulled apart. L_{SC} pushes together embeddings of points from the same class and pulls apart embeddings of points from different classes. In contrast, our modified

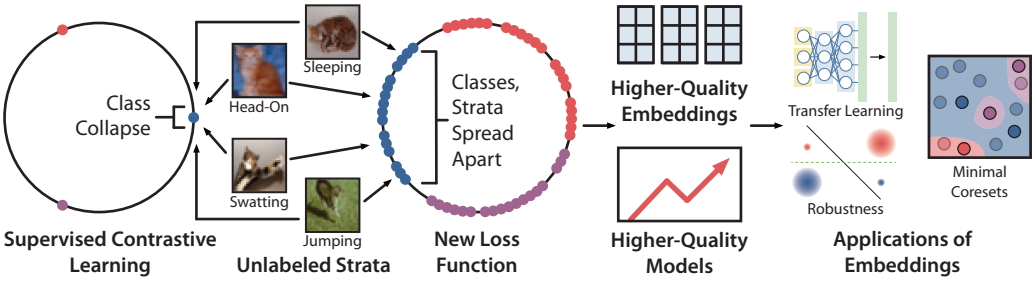

Figure 1: Classes contain critical information that is not explicitly encoded in the class labels. Supervised contrastive learning (left) loses this information, since it maps unlabeled strata such as sleeping cats, jumping cats, and swatting cat to a single embedding. We introduce a new loss function L_{spread} that prevents class collapse and maintains strata distinctions. L_{spread} produces higher-quality embeddings, which we evaluate with three downstream applications.

loss L_{spread} includes an additional class-conditional InfoNCE loss term that uniformly pulls apart individual points from within the same class. This term encourages points from the same class to be maximally spread apart in embedding space, which discourages class collapse (see Figure 1 middle). Surprisingly, even though L_{spread} does not use strata labels, it still produces embeddings that qualitatively appear to retain more strata information than those produced by L_{SC} (see Figure 2).

In Section 4, motivated by these empirical observations, we build off previous theoretical work (Graf et al., 2021; Wang & Isola, 2020) to study how well L_{spread} preserves distinctions between strata in the representation space. We propose a simple thought experiment considering the embeddings that the supervised contrastive loss generates when it is trained on a fraction of the dataset. This setup enables us to distinguish strata based on their sizes by considering how likely it is for them to be represented in the sample (larger strata are more likely to appear in a small sample). When strata do not appear in the sample, we view them as out-of-distribution data with embeddings characterized by their information-theoretic properties. As a result, we can show that the supervised contrastive loss has different effects on different strata depending on their size and distribution—and that L_{spread} increases the magnitude of these differences. Colloquially, rarer and more distinct strata are farther away from common strata, and we show that this property is important for embedding quality.

In Section 5, we empirically explore several downstream implications of these insights. We demonstrate that L_{spread} produces embeddings that retain more information about strata, which enables lift on a number of downstream applications that require strata recovery. We present three such downstream applications to evaluate embedding quality in this paper:

- We evaluate how well L_{spread}'s embeddings encode fine-grained subclasses with coarse-to-fine transfer learning. L_{spread} achieves up to 4.4 points of lift across four datasets.
- We evaluate how well embeddings produced by L_{spread} can recover strata in an unsupervised setting by evaluating robustness against worst-group accuracy and noisy labels. We use our insights about how L_{spread} embeds strata of different sizes to improve worst-group robustness by up to 2.5 points and to recover 75% performance when 20% of the labels are noisy.
- We evaluate how well we can differentiate rare strata from common strata by constructing limited subsets of the training data that can achieve the highest performance under a fixed training strategy (the coreset problem). We construct coresets by subsampling points from common strata. Our coresets outperform prior work by 1.0 points when coreset size is 30% of the training set.

In addition, we find that L_{spread} produces higher-quality models, outperforming L_{SC} by up to 4.0 points across 9 tasks. Finally, we discuss related work in Section 6 and conclude in Section 7.

2 Background

We present our generative model for strata (Section 2.1. Then, we discuss supervised contrastive learning—in particular the SupCon loss L_{SC} from Khosla et al. (2020) and its optimal embedding distribution Graf et al. (2021)—and the end model for classification (Section 2.2).

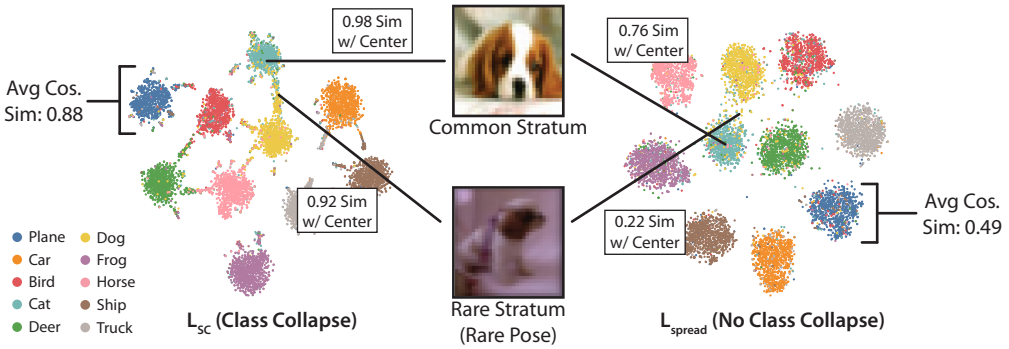

Figure 2: L_{spread} produces embeddings that are qualitatively better than those produced by L_{SC}. We show t-SNE visualizations of embeddings for the CIFAR10 test set and report cosine similarity metrics (average intracluster cosine similarities, and similarities between individual points and the class cluster). L_{spread} produces lower intraclass cosine similarity and embeds images from rare strata further out over the hypersphere than L_{SC}.

2.1 Data Setup

We have a labeled input dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, where $(x, y) \sim \mathcal{P}$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y} = \{1, \ldots, K\}$. For a particular data point x, we denote its label as $h(x) \in \mathcal{Y}$ with distribution $p(y|x)$. We assume that data is class-balanced such that $p(y = i) = \frac{1}{K}$ for all $i \in \mathcal{Y}$. The goal is to learn a model $\hat{p}(y|x)$ on $\mathcal{D}$ to classify points.

Data points also belong to categories beyond their labels, called *strata*. Following Sohoni et al. (2020), we denote a stratum as a latent variable z, which can take on values in $\mathcal{Z} = \{1, \ldots, C\}$. $\mathcal{Z}$ can be partitioned into disjoint subsets $S_1, \ldots, S_K$ such that if $z \in S_k$, then its corresponding y label is equal to k. Let $S(c)$ denote the deterministic label corresponding to stratum c. We model the data generating process as follows. First, the latent stratum is sampled from distribution $p(z)$. Then, the data point x is sampled from the distribution $\mathcal{P}_z = p(\cdot|z)$, and its corresponding label is $y = S(z)$ (see Figure 2 of Sohoni et al. (2020). We assume that each class has m strata, and that there exist at least two strata, z_1, z_2, where $S(z_1) \neq S(z_2)$ and $\text{supp}(z_1) \cap \text{supp}(z_2) \neq \emptyset$.

2.2 Supervised Contrastive Loss

Supervised contrastive loss pushes together pairs of points from the same class (called positives) and pulls apart pairs of points from different classes (called negatives) to train an encoder $f : \mathcal{X} \to \mathbb{R}^d$. Following previous works, we make three assumptions on the encoder: 1) we restrict the encoder output space to be $\mathbb{S}^{d-1}$, the unit hypersphere; 2) we assume $K \leq d + 1$, which allows Graf et al. (2021) to recover optimal embedding geometry; and 3) we assume the encoder f is infinitely powerful, meaning that any distribution on $\mathbb{S}^{d-1}$ is realizable by $f(x)$.

SupCon and Collapsed Embeddings We focus on the SupCon loss L_{SC} from Khosla et al. (2020). Denote $\sigma(x, x') = f(x)^\top f(x')/\tau$, where τ is a temperature hyperparameter. Let $\mathcal{B}$ be the set of batches of labeled data on $\mathcal{D}$ and $P(i, B) = \{p \in B \backslash i : h(p) = h(i)\}$ be the points in B with the same label as x_i. For an anchor x_i, the SupCon loss is $\hat{L}_{SC}(f, x_i, B) = \frac{-1}{|P(i,B)|} \sum_{p \in P(i,B)} \log \frac{\exp(\sigma(x_i, x_p))}{\sum_{a \in B \backslash i} \exp(\sigma(x_i, x_a))}$, where $P(i, B)$ forms positive pairs and $B \backslash i$ forms negative pairs.

The optimal embedding distribution that minimizes L_{SC} contains one embedding per class, with the per-class embeddings collectively forming a regular simplex inscribed in the hypersphere Graf et al. (2021). Formally, if $h(x) = i$, then $f(x) = v_i$ for all $x \in \mathcal{B}$. $\{v_i\}_{i=1}^K$ makes up the regular simplex, defined by: a) $\sum_{i=1}^K v_i = 0$; b) $\|v_i\|_2 = 1$; and c) $\exists c_K \in \mathbb{R}$ s.t. $v_i^\top v_j = c_K$ for $i \neq j$. We describe this property as *class collapse* and define the distribution of $f(x)$ that satisfies these conditions as *collapsed embeddings*.

End Model After the supervise contrastive loss is used to train an encoder, a linear classifier $W \in \mathbb{R}^{K \times d}$ is trained on top of the representations $f(x)$ by minimizing cross-entropy loss over softmax scores. We assume that $\|W_y\|_2 \leq 1$ for each $y \in \mathcal{Y}$. The end model's empirical loss can be defined as $\hat{\mathcal{L}}(W, \mathcal{D}) = \sum_{x_i \in \mathcal{D}} -\log \frac{\exp(f(x_i)^\top W_{h(x_i)})}{\sum_{j=1}^{K} \exp(f(x_i)^\top W_j)}$. The model uses softmax scores constructed with $f(x)$ and W to generate predictions $\hat{p}(y|x) = \hat{p}(y|f(x))$. Finally, the generalization error of the model on $\mathcal{P}$ is the expected cross-entropy between $\hat{p}(y|x)$ and $p(y|x)$, namely $\mathcal{L}(x, y, f) = \mathbb{E}_{x,y}\left[-\log \hat{p}(y|f(x))\right]$.

3 METHOD

We now highlight some theoretical problems with class collapse under our generative model of strata (Section 3.1). We then propose and qualitatively analyze a loss function L_{spread} (Section 3.2).

3.1 THEORETICAL MOTIVATION

We describe conditions when collapsed embeddings minimize generalization error on coarse-to-fine transfer and the original task. We find that these conditions do not hold when distinct strata exist.

Consider the downstream *coarse-to-fine transfer* task (x, z) of using embeddings $f(x)$ learned on (x, y) to classify points by fine-grained strata. Formally, coarse-to-fine transfer involves learning an end model with weight matrix $W \in \mathbb{R}^{C \times d}$ and fixed $f(x)$ (as described in Section 2.2) on points (x, z), where we assume the data are class-balanced across z.

Observation 1. *Class collapse minimizes $\mathcal{L}(x, z, f)$ if for all x, 1) $p(y = h(x)|x) = 1$, meaning that each x is deterministically assigned to one class, and 2) $p(z|x) = \frac{1}{m}$ where $z \in S_{h(x)}$. The second condition implies that $p(x|z) = p(x|y)$ for all $z \in S_y$, meaning that there is no distinction among strata from the same class. This contradicts our generative model assumptions.*

Similarly, we characterize when collapsed embeddings are optimal for the original task (x, y).

Observation 2. *Class collapse minimizes $\mathcal{L}(x, y, f)$ if, for all x, $p(y = h(x)|x) = 1$. This contradicts our generative model assumptions.*

Proofs are in Appendix D.1. We also show in Appendix C.1 that a one-to-one encoder obeys the Infomax principle (Linsker, 1988) better than collapsed embeddings on new distributions (x', y'). These observations suggest that a distribution over the embeddings that preserves strata distinctions and does not collapse classes is more desirable.

3.2 MODIFIED CONTRASTIVE LOSS L_{spread}

We introduce the loss L_{spread}, a weighted sum of two contrastive losses $L_{attract}$ and L_{repel}. $L_{attract}$ is a supervised contrastive loss, while L_{repel} encourages intra-class separation. For $\alpha \in [0, 1]$,

$$L_{spread} = \alpha L_{attract} + (1 - \alpha) L_{repel}. \tag{1}$$

For a given anchor x_i, define x_i^{aug} as an augmentation of the same point as x. Define the set of negative examples for i to be $N(i, B) = \{a \in B \backslash i : h(a) \neq h(i)\}$. Then,

$$\hat{L}_{attract}(f, x_i, B) = \frac{-1}{|P(i, B)|} \sum_{p \in P(i,B)} \log \frac{\exp(\sigma(x_i, x_p))}{\exp(\sigma(x_i, x_p)) + \sum_{a \in N(i,B)} \exp(\sigma(x_i, x_a))} \tag{2}$$

$$\hat{L}_{repel}(f, x_i, B) = -\log \frac{\exp(\sigma(x_i, x_i^{aug}))}{\sum_{p \in P(i,B)} \exp(\sigma(x_i, x_p))} \tag{3}$$

$\hat{L}_{attract}$ is a variant of the SupCon loss, which encourages class separation in embedding space as suggested by Graf et al. (2021). $\hat{L}_{repel}$ is a class-conditional InfoNCE loss, where the positive distribution consists of augmentations and the negative distribution consists of i.i.d samples from the same class. It encourages points within a class to be spread apart, as suggested by the analysis of the InfoNCE loss by Wang & Isola (2020).

Qualitative evaluation Figure 2 shows t-SNE plots for embeddings produced with L_{SC} versus L_{spread} on the CIFAR10 test set. L_{spread} produces embeddings that are more spread out than those produced by L_{SC} and avoids class collapse. As a result, images from different strata can be better differentiated in embedding space. For example, we show two dogs, one from a common stratum and one from a rare stratum (rare pose). The two dogs are much more distinguishable by distance in the L_{spread} embedding space, which suggests that it helps preserve distinctions between strata.

4 Theoretical Analysis

In this section, we first apply current theoretical tools to L_{spread} to understand the optimal embedding distributions. These tools do not fully capture strata behavior. In Section 4.1, we propose a simple thought experiment about the distances between strata in embedding space when trained under a finite subsample of data to explain our prior qualitative observations. Then, in Section 4.2, we analyze how L_{spread} produces better representations for both coarse-to-fine transfer and the original task (x, y).

Existing Analysis Previous works have studied the geometry of optimal embeddings under contrastive learning (Graf et al., 2021; Wang & Isola, 2020; Robinson et al., 2020), but their techniques cannot analyze strata. As an example, we adopt and expand the analysis from Wang & Isola (2020). First, we set up some notation to analyze L_{spread} asymptotically. For an anchor x, the positive example x^+ is drawn from $p^+(\cdot|x) = p(\cdot|h(x^+) = h(x))$. Negative examples x^- are drawn from $p^-(\cdot|x) = p(\cdot|h(x^-))$. An augmentation x^{aug} is drawn from $p_a(\cdot|x)$.

Theorem 1. *Define $n^+ = |P(i, B)|, n^- = |N(i, B)|$. As $n^+, n^- \to \infty$, the population-level loss over batches and anchors, which we denote as $L_{spread}(f, n^+, n^-)$ (see Definition 2), converges to:*

$$\lim_{n^+, n^- \to \infty} L_{spread}(f, n^+, n^-) - (1-\alpha)\log n^+ - \alpha \log n^- = L_{align}(f) + L_{uniform}(f) + L_{neg}(f),$$

where

1. $L_{align}(f) = -\big(\alpha \mathbb{E}_{x,x^+\sim p^+}\left[\sigma(x, x^+)\right] + (1-\alpha)\mathbb{E}_{x,x^{aug}\sim p_a}\left[\sigma(x, x^{aug})\right]\big)$ *is minimized when all points from a class collapse to one embedding.*
2. $L_{uniform}(f) = (1-\alpha)\mathbb{E}_{x\sim\mathcal{P}}\left[\log \mathbb{E}_{x^+\sim p^+(\cdot|x)}\left[\exp(\sigma(x, x^+))\right]\right]$ *is minimized when points in each class are distributed uniformly on the hypersphere.*
3. $L_{neg}(f) = \alpha \mathbb{E}_{x\sim\mathcal{P}}\left[\log \mathbb{E}_{x^-\sim p^-(\cdot|x)}\left[\exp(\sigma(x, x^-))\right]\right]$ *is minimized when points from a class collapse to one embedding and collectively form a regular simplex inscribed in the hypersphere.*

The proof is in Appendix D.2. While the optimal distributions for $L_{uniform}$ and L_{neg} suggest better spread both among and within classes, several issues prohibit a closer analysis of L_{spread} for strata. First, the individual minima of $L_{uniform}$ and L_{neg} do not intersect, which prevents us from characterizing L_{spread}'s optimal distribution. Second, even if a unique optimal distribution were deduced in this way, it would not capture strata. This is because this approach models embeddings not as a mapping from $\mathcal{X}$, but as a distribution on the hypersphere based on information in the loss function. However, strata are unknown at training time and thus impossible to incorporate explicitly into the loss. Therefore, we need another explanation for our empirical observations that strata distinctions are preserved in embedding space under L_{spread}.

4.1 Geometry of Strata under Supervised Contrastive Loss

We propose a simple thought experiment based on *subsampling the dataset*—randomly sampling a fraction of the training data—to analyze strata. Consider the following: we subsample a fraction $t \in [0, 1]$ of a training set of N points from $\mathcal{P}$. We use this subsampled dataset $\mathcal{D}_t$ to learn an encoder $\hat{f}_t$, and we study the average distance under $\hat{f}_t$ between two strata z and z' from the same class as t varies.

The average distance between z and z' is $\delta(\hat{f}_t, z, z') = \|\mathbb{E}_{x\sim\mathcal{P}_z}[\hat{f}_t(x)] - \mathbb{E}_{x\sim\mathcal{P}_{z'}}[\hat{f}_t(x)]\|_2$ and depends on whether z and z' are both in the subsampled dataset. We have three cases (with probabilities stated in Appendix C.2) based on strata frequency and t—when both, one, or neither of the strata appears in $\mathcal{D}_t$:

1. **Both strata appear in** $\mathcal{D}_t$ The encoder $\hat{f}_t$ is trained on both z and z'. For large N, we can approximate the setting by considering $\hat{f}_t$ trained on infinite data from these strata. The optimal embedding distribution from Theorem 1 is defined on these strata, so $\delta(\hat{f}_t, z, z')$ can be analyzed by examining properties of that distribution. Note that if L_{SC} were used, $\delta(\hat{f}_t, z, z')$ converges to 0. This case occurs with probability increasing in $p(z), p(z')$, and t.
2. **One stratum but not the other appears in** $\mathcal{D}_t$ Without loss of generality, suppose that points from z appear in $\mathcal{D}_t$ but no points from z' do. Since the downstream classifier $\hat{p}(y|\hat{f}_t(x))$ is a function of distances in embedding space, we can equivalently consider how the end model learned using the "source" distribution containing z performs on the "target" distribution of stratum z'. Borrowing from literature in domain adaptation, the difficulty of this out-of-distribution problem depends on both the divergence between source and target distributions and the capacity of the overall model. For instance, the $\mathcal{H}\Delta\mathcal{H}$-divergence from Ben-David et al. (2010; 2007), which is studied in lower bounds in Ben-David & Urner (2012), and the discrepancy difference from Mansour et al. (2009) capture both concepts. Moreover, L_{spread} and L_{SC} induce different end model hypothesis classes, which can help explain why L_{spread} better preserves strata distances. This case occurs with probability increasing in $p(z)$ and decreasing in $p(z')$ and t.
3. **Neither strata appears in** $\mathcal{D}_t$ The distance $\delta(\hat{f}_t, z, z')$ is at most $2D_{\text{TV}}(\mathcal{P}_z, \mathcal{P}_{z'})$ (total variation distance), regardless of how the encoder is trained. This case occurs with probability decreasing in $p(z), p(z')$, and t.

We make two observations from these cases. First, if z and z' are both common strata, then as t increases, the distance between them depends on the optimal asymptotic distribution. Therefore, if we set $\alpha = 1$ in L_{spread}, these common strata will collapse. Second, if z is a common strata and z' is uncommon, the second case occurs frequently over randomly sampled $\mathcal{D}_t$, and thus the strata are separated based on the difficulty of the respective out-of-distribution problem. We thus arrive at the following insight from our thought experiment:

> *Common strata are more tightly clustered together, while rarer and more semantically distinct strata are far away from them.*

Figure 3 demonstrates this insight. Points from the largest subclasses (dark blue) cluster tightly, whereas points from small subclasses (light blue) are scattered throughout the embedding space.

4.2 Implications

We discuss theoretical and practical implications of our subsampling argument. First, we show that on both the coarse-to-fine transfer task (x, z) and the original task (x, y), embeddings that preserve strata yield better generalization error. Second, we discuss practical implications arising from our subsampling argument that enable new applications.

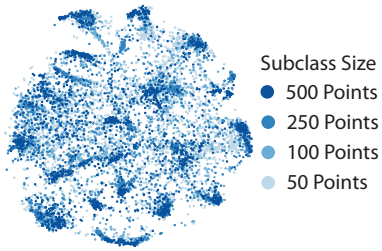

Figure 3: Points from large subclasses cluster tightly; points from small subclasses scatter (CIFAR100-Coarse, unbalanced subclasses).

Theoretical Implications Consider $\hat{f}_1$, the encoder trained on $\mathcal{D}$ with N points using L_{spread}, and suppose a mean classifier is used for the end model. On coarse-to-fine transfer, generalization error depends on how far each stratum center is from the others.

Lemma 1. *The generalization error on the coarse-to-fine transfer task is at most* $\mathcal{L}(x, z, \hat{f}_1) \leq \mathbb{E}_z\Big[\log\big(\sum_{i:S(i)=S(z)} \exp(-\delta_{intra}(z, i)) + \sum_{i:S(i)\neq S(z)} \exp(-\delta_{inter}(z, i))\big)\Big] - 1$, *where* $\delta_{intra}(z, i)$ *and* $\delta_{inter}(z, i)$ *are quantities that scale with the distances between strata* z *and* i *depending on if* $S(z) = S(i)$ *(see Appendix D.3 for exact expressions). Note that* $\delta_{intra}(z, i)$ *corresponds with the distances considered in our thought experiment in Section 4.1.*

The larger the distances between strata, the smaller the upper bound on generalization error. A similar result holds on the original task (x, y), but there is an additional term that penalizes points from the same class being too far apart.

Dataset	End Model Perf. L_{SS}	L_{SC}	L_{spread}
CIFAR10	89.7	90.9	**91.5**
CIFAR10-Coarse	97.7	96.5	**98.1**
CIFAR100	68.0	67.5	**69.1**
CIFAR100-Coarse	76.9	77.2	**78.3**
CIFAR100-Coarse-U	72.1	71.6	**72.4**
MNIST	99.1	**99.3**	99.2
MNIST-Coarse	99.1	**99.4**	**99.4**
Waterbirds	77.8	73.9	**77.9**
ISIC	87.8	88.7	**90.0**

Dataset	Coarse-to-Fine Transfer L_{SS}	L_{SC}	L_{spread}
CIFAR10-Coarse	71.7	52.5	**76.1**
CIFAR100-Coarse	62.0	62.4	**63.9**
CIFAR100-Coarse-U	61.9	59.5	**62.4**
MNIST-Coarse	97.1	98.8	**99.0**

Figure 4: **Left:** 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. **Right:** Performance of coarse-to-fine transfer on various datasets compared against contrastive baselines. In these tasks, we first train a model on coarse task labels, then freeze the representation and train a model on fine-grained subclass labels. L_{spread} produces embeddings that transfer better across all datasets.

Lemma 2. *The generalization error on the original task is at most* $\mathcal{L}(x, y, \hat{f}_1) \leq \mathbb{E}_z\Big[\log\big(\exp(-\delta_{intra}(z)) + \sum_{i \neq S(z)} \exp(-\delta_{inter}(z))\big) + \frac{1}{\lambda_y}\delta_{intra}(z)\Big]$. $\lambda > 0$ *is a constant.* $\delta_{intra}(z)$ *scales with the average distances between* z *and other strata in its class, which we analyze in Section 4.1.* $\delta_{inter}(z)$ *scales with the average distances between* z *and strata in other classes. See Appendix D.3 for exact expressions.*

Practical Implications Our discussion in Section 4.1 suggests that training with L_{spread} better distinguishes strata in embedding space. As a result, we can use differences between strata of different sizes for downstream applications. For example, unsupervised clustering can help recover pseudolabels for unlabeled, rare strata. These pseudolabels can be used as inputs to worst-group robustness algorithms, or used to detect noisy labels, which appear to be rare strata during training (see Section 5.2 for examples). We can also train over subsampled datasets to heuristically distinguish points that come from common strata from points that come from rare strata. We can then downsample points from common strata to construct minimal coresets (see Section 5.3 for examples).

5 Experiments

This section evaluates L_{spread} on embedding quality and model quality:

- First, in Section 5.1, we use coarse-to-fine transfer learning to evaluate how well the embeddings maintain strata information. We find that L_{spread} achieves lift across four datasets.
- In Section 5.2, we evaluate how well L_{spread} can detect rare strata in an unsupervised setting. We first use L_{spread} to detect rare strata to improve worst-group robustness by up to 2.5 points. We then use rare strata detection to correct noisy labels, recovering 75% performance under 20% noise.
- In Section 5.3, we evaluate how well L_{spread} can distinguish points from large strata versus points from small strata. We downsample points from large strata to construct minimal coresets on CIFAR10, outperforming prior work by 1.0 points at 30% labeled data.
- Finally, in Section 5.4, we show that training with L_{spread} improves model quality, validating our theoretical claims that preventing class collapse can improve generalization error. We find that L_{spread} improves performance in 7 out of 9 cases.

Datasets and Models Figure 4 (left) lists all our datasets. CIFAR10, CIFAR100, and MNIST are the standard computer vision datasets. We also use coarse versions of each, wherein classes are combined to create coarse superclasses (animals/vehicles for CIFAR10, standard superclasses for CIFAR100, and <5, ≥ 5 for MNIST). In CIFAR100-Coarse-U, some subclasses have been artificially imbalanced. Waterbirds and ISIC are real-world datasets with documented hidden strata (Sagawa et al., 2019; Codella et al., 2019; Sohoni et al., 2020). We use a ViT model (Dosovitskiy et al., 2020)

Dataset	Sub-Group Recovery L_{CE}	L_{SC}	L_{spread}
Waterbirds	56.3	47.2	**59.0**
ISIC	74.0	92.5	**93.8**
	Worst-Group Robustness		
Waterbirds	88.4	86.5	**89.0**
ISIC	92.0	**93.3**	92.6

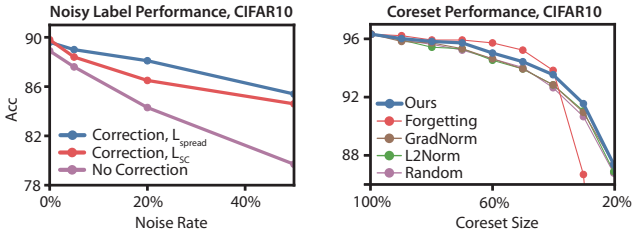

Figure 5: **Left:** Unsupervised strata recovery performance (top, F1), and worst-group performance (AUROC for ISIC, Acc for others) using recovered strata. **Center:** Performance of models under various amounts of label noise for the contrastive loss head. **Right:** Performance of a ResNet18 trained with coresets of various sizes.

(4x4, 7 layers) for CIFAR and MNIST and a ResNet50 for the rest. 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. More details in Appendix E.

5.1 Coarse-to-Fine Transfer Learning

In this section, we use coarse-to-fine transfer learning to evaluate how well L_{spread} retains strata information in the embedding space. We train on coarse superclass labels, freeze the weights, and then use transfer learning to train a linear layer with subclass labels. We use this supervised strata recovery setting to isolate how well the embeddings can recover strata in the optimal setting.

Figure 4 (right) reports the results. We find that L_{spread} produces better embeddings for coarse-to-fine transfer learning than L_{SC} and L_{SS}. Lift over L_{SC} varies from 0.2 points on MNIST (16.7% error reduction), to 23.6 points of lift on CIFAR10. L_{spread} also produces better embeddings than L_{SS}, since L_{SS} does not encode superclass labels in the embedding space.

5.2 Robustness Against Worst-Group Accuracy and Noise

In this section, we use robustness to measure how well L_{spread} can recover strata in an unsupervised setting. We use clustering to detect rare strata as an input to worst-group robustness algorithms, and we use a geometric heuristic over embeddings to correct noisy labels.

To evaluate worst-group accuracy, we follow the experimental setup and datasets from Sohoni et al. (2020). We first train a model with class labels. We then cluster the embeddings to produce pseudolabels for hidden strata, which we use as input for a Group-DRO algorithm to optimize worst-group robustness (Sagawa et al., 2019). To evaluate robustness against noise, we introduce noisy labels to the contrastive loss head on CIFAR10. We detect noisy labels with a simple geometric heuristic: points with incorrect labels appear to be small strata, so they should be far away from other points of the same class. We then correct noisy points by assigning the label of the nearest cluster in the batch. More details can be found in Appendix E.

Figure 5 (left) shows the performance of unsupervised strata recovery and downstream worst-group robustness. We can see that L_{spread} outperforms both L_{SC} and L_{CE} on strata recovery. This translates to better worst-group robustness on the Waterbirds task, outperforming L_{SC} by 2.5 points, and L_{CE} by 0.6 points.

Figure 5 (center) shows the effect of noisy labels on performance. When noisy labels are uncorrected (purple), performance drops by up to 10 points at 50% noise. Applying our geometric heuristic (red) can recover 4.8 points at 50% noise, even without using L_{spread}. But L_{spread} recovers an additional 0.9 points at 50% noise, and an additional 1.6 points at 20% noise (blue). In total, L_{spread} recovers 75% performance at 20% noise, whereas L_{SC} only recovers 45% performance.

5.3 Minimal Coreset Construction

Now we evaluate how well training on fractional samples of the dataset with L_{spread} can distinguish points from large versus small strata by constructing minimal coresets for CIFAR10. We train a

ResNet18 on CIFAR10, following Toneva et al. (2019), and compare against baselines from Toneva et al. (2019) and Paul et al. (2021). For our coresets, we train with L_{spread} on subsamples of the dataset and record how often points are correctly classified at the end of each run. We bucket points in the training set by how often the point is correctly classified. We then iteratively remove points from the largest bucket in each class. Our strategy removes easy examples first from the largest coresets, but maintains a set of easy examples in the smallest coresets.

Figure 5 (right) shows the results at various coreset sizes. For large coresets, our algorithm outperforms both methods from Paul et al. (2021) and is competitive with Toneva et al. (2019). For small coresets, our method outperforms the baselines, providing up to 5.2 points of lift over Toneva et al. (2019) at 30% labeled data. Our analysis helps explain this gap; removing too many easy examples hurts performance, since then the easy examples become rare and hard to classify.

5.4 Model Quality

Finally, we confirm that L_{spread} produces higher-quality models and achieves better sample complexity than both L_{SC} and the SimCLR loss L_{SS}. Figure 4 (left) reports the performance of models across all our datasets. We find that L_{spread} achieves better overall performance compared to models trained with L_{SC} and L_{SS} in 7 out of 9 tasks, and matches performance in 1 task. We find up to 4.0 points of lift over L_{SC} (Waterbirds), and up to 2.2 points of lift (AUROC) over L_{SS} (ISIC). In Appendix F, we additionally evaluate the sample complexity of contrastive losses by training on partial subsamples of CIFAR10. L_{spread} outperforms L_{SC} and L_{SS} throughout.

6 Related Work and Discussion

From work in **contrastive learning**, we most directly extend Wang & Isola (2020) and Graf et al. (2021), who study representations on the hypersphere along with Robinson et al. (2020). 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. We initially started from a debiasing angle to study the effects of noise in supervised contrastive learning inspired by Chuang et al. (2020), but moved to our current strata-based view of noise instead. 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), so we focus on the geometric perspective in this paper because of the downstream applications. 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).

Our treatment of **strata** 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).

Each of our downstream **applications** is a field in itself, and we take inspiration from recent work from each. Our noise heuristic is similar to the ELR (Liu et al., 2020) and takes inspiration from a various work using contrastive learning to correct noisy labels and for semi-supervised learning (Li et al., 2021; Ciortan et al., 2021; Li et al., 2020). Our coreset algorithm is inspired by recent work in coresets for modern deep networks (Ju et al., 2021; Sener & Savarese, 2018; Paul et al., 2021), and takes inspiration from Toneva et al. (2019) in particular.

7 Conclusion

We propose a new supervised contrastive loss function to prevent class collisions and produce higher-quality embeddings. We show that our loss function maintains strata distinctions in embedding space and explore several downstream applications. Future directions include encoding hierarchies in the contrastive loss functions and extending our work to more modalities, models, and applications. We hope that our work inspires further work in more fine-grained supervised contrastive loss functions and new theoretical approaches for reasoning about generalization and strata.

Reproducibility Statement For theoretical results, the main assumptions are listed in Section 2. A glossary of terms is provided in Appendix A, definitions are provided in Appendix B, and proofs are provided in Appendix D. Full experimental details about datasets and models are given in Appendix E. Code will be publicly released before publication.

Ethics Statement We hope that our work encourages the community to consider strata as a tool to analyze and evaluate the generalization of machine learning methods from a different perspective. We hope that our methods and analysis inspire future work in using contrastive learning to improve the robustness of machine learning models, especially when understanding the actions of different approaches on strata is important for safety or fairness.

REFERENCES

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

Shai Ben-David and Ruth Urner. On the hardness of domain adaptation and the utility of unlabeled target samples. In Nader H. Bshouty, Gilles Stoltz, Nicolas Vayatis, and Thomas Zeugmann (eds.), *Algorithmic Learning Theory*, pp. 139–153, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-34106-9.

Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 2020a.

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

Ching-Yao Chuang, Joshua Robinson, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. volume 33, 2020.

Madalina Ciortan, Romain Dupuis, and Thomas Peel. A framework using contrastive learning for classification with noisy labels, 2021.

Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

Greg d'Eon, Jason d'Eon, James R Wright, and Kevin Leyton-Brown. The spotlight: A general method for discovering systematic errors in deep learning models. *arXiv preprint arXiv:2107.00758*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

John Duchi, Tatsunori Hashimoto, and Hongseok Namkoong. Distributionally robust losses for latent covariate mixtures. *arXiv preprint arXiv:2007.13982*, 2020.

Karan Goel, Albert Gu, Yixuan Li, and Christopher Re. Model patching: Closing the subgroup performance gap with data augmentation. In *International Conference on Learning Representations*, 2020.

Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*, 2021.

Florian Graf, Christoph Hofer, Marc Niethammer, and Roland Kwitt. Dissecting supervised constrastive learning. In *International Conference on Machine Learning*, pp. 3821–3830. PMLR, 2021.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Achim Hoffmann, Rex Kwok, and Paul Compton. Using subclasses to improve classification learning. In *European Conference on Machine Learning*, pp. 203–213. Springer, 2001.

Jeongwoo Ju, Heechul Jung, Yoonju Oh, and Junmo Kim. Extending contrastive learning to unsupervised coreset selection. *arXiv preprint arXiv:2103.03574*, 2021.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Mschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. volume 33, 2020.

Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020.

Junnan Li, Caiming Xiong, and Steven C.H. Hoi. Semi-supervised learning with contrastive graph regularization. In *ICCV*, 2021.

R. Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988. doi: 10.1109/2.36.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, 33, 2020.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29), 2018.

Luke Oakden-Rayner, Jared Dunnmon, Gustavo Carneiro, and Christopher Ré. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In *Proceedings of the ACM conference on health, inference, and learning*, pp. 151–159, 2020.

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

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *arXiv preprint arXiv:2107.07075*, 2021.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2019.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33, 2020.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*, 2020.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Self-supervised learning from a multi-view perspective. In *International Conference on Learning Representations*, 2020.

Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *International Conference on Learning Representations*, 2020.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2016.

Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *Advances in Neural Information Processing Systems*, 27:487–495, 2014.

Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. *arXiv preprint arXiv:2012.08850*, 2021.

We provide a glossary in Appendix A. Then we provide definitions of terms in Appendix B. We discuss additional theoretical results in Appendix C. We provide proofs in Appendix D. We discuss additional experimental details in Appendix E. Finally, we provide additional experimental results in Appendix F.

A GLOSSARY

The glossary is given in Table 1 below.

Symbol	Used for
L_{SC}	SupCon (see Section 2.2), a supervised contrastive loss introduced by Khosla et al. (2020).
L_{spread}	Our modified loss function. It is defined empirically in Section 3.2 and theoretically in 2.
x	Input data $x \in \mathcal{X}$.
y	Class label $y \in \mathcal{Y} = \{1, \ldots, K\}$.
$\mathcal{D}$	Dataset of N points $\{(x_i, y_i)\}_{i=1}^N$ drawn i.i.d. from $\mathcal{P}$.
$h(x)$	The class that x belongs to, i.e. $h(x)$ is a label drawm from $p(y\|x)$. This label information is used as input in the supervised contrastive loss.
$\hat{p}(y\|x)$	The end model's predicted distribution over y given x.
z	A stratum is a latent variable $z \in \mathcal{Z} = \{1, \ldots, C\}$ that further categorizes data beyond labels.
S_k	The set of all strata corresponding to label k (deterministic).
$S(c)$	The label corresponding to strata c (deterministic).
$\mathcal{P}_z$	The distribution of input data belonging to stratum z, i.e. $x \sim p(\cdot\|z)$.
m	The number of strata per class.
d	Dimension of the embedding space.
f	The encoder $f : \mathcal{X} \to \mathbb{R}^d$ maps input data to an embedding space and is learned by minimizing the contrastive loss function.
$\mathbb{S}^{d-1}$	The unit hypersphere, formally $\{v \in \mathbb{R}^d : \|v\|_2 = 1\}$.
τ	Temperature hyperparameter in contrastive loss function.
$\sigma(x, x')$	Notation for $\frac{f(x)^\top f(x')}{\tau}$.
$\mathcal{B}$	Set of batches of labeled data on $\mathcal{D}$.
$P(i, B)$	Points in B with the same label as x_i, formally $\{p \in B \backslash i : h(p) = h(i)\}$.
$\{v_i\}_{i=1}^K$	A regular simplex inscribed in the hypersphere (see Definition 1).
W	The weight matrix that parametrizes the downstream linear classifier (end model) learned on $f(x)$.
$\hat{\mathcal{L}}(W, \mathcal{D})$	The empirical cross entropy loss used to learn W over dataset $\mathcal{D}$ (see (7)).
$\mathcal{L}(x, y, f)$	The generalization error of the end model of predicting output y on x using encoder f (see (8) and (9)).
$L_{attract}$	A variant on SupCon that is used in L_{spread} that pushes points of a class together (see (2)).
L_{repel}	A class-conditional InfoNCE loss that is used in L_{spread} to pull apart points within a class (see (3)).
α	Hyperparameter $\alpha \in [0, 1]$ controls how to balance $L_{attract}$ and L_{repel}.
x^{aug}	An augmentation of data point x.
$N(i, B)$	Points in B with a label different from that of x_i, formally $\{a \in B \backslash i : h(a) \neq h(i)\}$.
$p^+(\cdot\|x)$	The distribution for positive example x^+, $p(\cdot\|h(x^+) = h(x))$ given anchor x.
$p^-(\cdot\|x)$	The distribution for negative example x^-, $p(\cdot\|h(x^-) \neq h(x))$ given anchor x.
$p_a(\cdot\|x)$	The distribution for augmented point x^{aug} given anchor x.
t	Fraction of training data $t \in [0, 1]$ that is varied in our thought experiment.
$\mathcal{D}_t$	Randomly sampled dataset from $\mathcal{P}$ with size equal to $t \cdot N$ fraction of $\mathcal{D}$.
$\hat{f}_t$	Encoder trained on sampled dataset $\mathcal{D}_t$.
$\delta(\hat{f}_t, z, z')$	The distance between centers of strata z and z' under encoder $\hat{f}_t$, namely $\delta(\hat{f}_t, z, z') = \|\mathbb{E}_{x \sim \mathcal{P}_z}[\hat{f}_t(x)] - \mathbb{E}_{x \sim \mathcal{P}_{z'}}[\hat{f}_t(x)]\|_2$.

Table 1: Glossary of variables and symbols used in this paper.

B DEFINITIONS

We restate definitions used in our proofs.

Definition 1 (Regular Simplex)**.** *The points $\{v_i\}_{i=1}^K$ form a regular simplex inscribed in the hypersphere if*

1. $\sum_{i=1}^K v_i = 0$
2. $\|v_i\| = 1$ *for all* i
3. $\exists c_K \leq 1$ *s.t.* $v_i^\top v_j = c_K$ *for* $i \neq j$

Definition 2 (L_{spread})**.** *We present the population-level version of L_{spread} defined in Section 3.2, $L_{spread}(f, n^+, n^-)$. Recall that $\sigma(x, x') = f(x)^\top f(x')/\tau$.*

$$L_{spread}(f, n^+, n^-) = \alpha L_{attract}(f, n^-) + (1-\alpha) L_{repel}(f, n^+), \tag{4}$$

where

$$L_{attract}(f, n^-) = \mathbb{E}_{\substack{x\sim\mathcal{P}, \\ x^+\sim p^+(\cdot|x), \\ \{x_i^-\}_{i=1}^{n^-}\sim p^-(\cdot|x)}} \left[-\log \frac{\exp(\sigma(x, x^+))}{\exp(\sigma(x, x^+)) + \sum_{i=1}^{n^-} \exp(\sigma(x, x_i^-))} \right], \tag{5}$$

$$L_{repel}(f, n^+) = \mathbb{E}_{\substack{x\sim\mathcal{P}, \\ x^{aug}\sim p_a(\cdot|x), \\ \{x_i^+\}_{i=1}^{n^+}\sim p^+(\cdot|x)}} \left[-\log \frac{\exp(\sigma(x, x^{aug}))}{\exp(\sigma(x, x^{aug})) + \sum_{i=1}^{n^+} \exp(\sigma(x, x_i^+))} \right]. \tag{6}$$

Definition 3 (Downstream model)**.** *Once an encoder $f(x)$ is learned, the downstream model consists of a linear classifier trained using the cross-entropy loss:*

$$\hat{\mathcal{L}}(W, \mathcal{D}) = \sum_{x_i \in \mathcal{D}} -\log \frac{\exp(f(x_i)^\top W_{h(x_i)})}{\sum_{j=1}^K \exp(f(x_i)^\top W_j)}. \tag{7}$$

Define $\hat{W} := argmin_{\|W\|^2 \leq 1} \hat{\mathcal{L}}(W, \mathcal{D})$. Then, the end model's outputs are the probabilities

$$\hat{p}(y|x) = \hat{p}(y|f(x)) = \frac{\exp(f(x)^\top \hat{W}_y)}{\sum_{j=1}^K, \exp(f(x)^\top \hat{W}_j)} \tag{8}$$

and the generalization error is

$$\mathcal{L}(x, y, f) = \mathbb{E}_{x,y} \left[-\log \hat{p}(y|f(x)) \right]. \tag{9}$$

C Additional Theoretical Results

C.1 Transfer learning on (x', y')

We now show an additional transfer learning result on new tasks (x', y'). Formally, recall that we learn the encoder f on $(x, y) \sim \mathcal{P}$. We wish to use it on a new task with target distribution $(x', y') \sim \mathcal{P}'$. We find that an injective encoder $f(x)$ is more appropriate to be used on new distributions than collapsed embeddings based on the Infomax principle (Linsker, 1988).

Observation 3. *Define $f_c(y)$ as the mapping to collapsed embeddings and $f_{1-1}(x)$ as an injective mapping, both learned on $\mathcal{P}$. Construct a new variable $\widetilde{y}$ with joint distribution $(x', \widetilde{y}) \sim p(y|x) \cdot p'(x')$ and suppose that $\widetilde{y} \perp\!\!\!\perp y'|x'$. Then, by the data processing inequality, it holds that $I(\widetilde{y}, y') \leq I(x', y')$ where $I(\cdot, \cdot)$ is the mutual information between two random variables. We apply f_c to $\widetilde{y}$ and f_{1-1} to x' to get that*

$$I(f_c(\widetilde{y}), y') \leq I(f_{1-1}(x'), y')$$

Therefore, f_{1-1} obeys the Infomax principle (Linsker, 1988) better on $\mathcal{P}'$ than f_c. Via Fano's inequality, this statement implies that the Bayes risk for learning y' from x' is lower using f_{1-1} than f_c.

C.2 PROBABILITIES OF STRATA z, z' APPEARING IN SUBSAMPLED DATASET

As discussed in Section 4.1, the distance between strata z and z' in embedding space depends on if these strata appear in the subsampled dataset $\mathcal{D}_t$ that the encoder was trained on. We define the exact probabilities of the three cases presented. Let $\Pr(z, z' \in \mathcal{D}_t)$ be the probability that both strata are seen, $\Pr(z \in \mathcal{D}_t, z' \notin \mathcal{D}_t)$ be the probability that only z is seen, and $\Pr(z, z' \notin \mathcal{D}_t$ be the probability that neither are seen.

First, the probability of neither strata appearing in $\mathcal{D}_t$ is easy to compute. In particular, we have that $\Pr(z, z' \notin \mathcal{D}_t) = (1 - p(z) - p(z'))^{tN}$.

Second, the probability of z being in $\mathcal{D}_t$ and z' not being in $\mathcal{D}_t$ can be expressed as $\Pr(z \in \mathcal{D}_t | z' \notin \mathcal{D}_t) \cdot \Pr(z' \notin \mathcal{D}_t)$. $\Pr(z' \notin \mathcal{D}_t)$ is equal to $(1 - p(z'))^{tN}$, and $\Pr(z \in \mathcal{D}_t | z' \notin \mathcal{D}_t) = 1 - \Pr(z \notin \mathcal{D}_t | z' \notin \mathcal{D}_t) = 1 - (1 - p(z | z \in \mathcal{Z} \backslash z'))^{tN}$. Finally, note that $p(z | z \in \mathcal{Z} \backslash z') = \frac{p(z)}{1 - p(z')}$. Putting this together, we get that $\Pr(z \in \mathcal{D}_t, z' \notin \mathcal{D}_t) = (1 - p(z'))^{tN} - (1 - p(z') - p(z))^{tN}$, and we can similarly construct $\Pr(z' \in \mathcal{D}_t, z \notin \mathcal{D}_t)$.

Lastly, the probability of both z and z' being in $\mathcal{D}_t$ is thus $\Pr(z, z' \in \mathcal{D}_t) = 1 - \Pr(z, z' \notin \mathcal{D}_t) - \Pr(z' \in \mathcal{D}_t, z \notin \mathcal{D}_t) - \Pr(z \in \mathcal{D}_t, z' \notin \mathcal{D}_t) = 1 + (1 - p(z') - p(z))^{tN} - (1 - p(z'))^{tN} - (1 - p(z))^{tN}$.

C.3 PERFORMANCE OF COLLAPSED EMBEDDINGS ON COARSE-TO-FINE TRANSFER AND ORIGINAL TASK

Lemma 3. *Denote f_c to be the encoder that collapses embeddings. Then, the generalization error on the coarse-to-fine transfer task using f_c and a linear classifier learned using cross entropy loss is at least*

$$\mathcal{L}(x, z, f_c) \geq \log(m \exp(1) + (C - m) \exp(c_K) - 1$$

where c_K is the dot product of any two different class-collapsed embeddings. The generalization error on the original task under the same setup is at least

$$\mathcal{L}(x, y, f_c) \geq \log(\exp(1) + (K - 1) \exp(c_K)) - 1$$

Proof. We first bound generalization error on the coarse-to-fine transfer task. For collapsed embeddings, $f(x) = v_i$ when $h(x) = i$, where $h(x)$ is information available at training time that follows the distribution $p(y|x)$. We thus denote the embedding $f(x)$ as $v_{h(x)}$. Therefore, we write the generalization error with an expectation over $h(x)$ and factorize the expectation according to our generative model.

$$\begin{aligned}
\mathbb{E}_{x,z,h(x)} \left[-\log \hat{p}(z | f(x)) \right] &= -\sum_{z=1}^{C} \sum_{h(x)=1}^{K} \int p(x, z, h(x)) \log \hat{p}(z | h(x)) dx \\
&= -\sum_{z=1}^{C} \sum_{h(x)=1}^{K} \int p(z) p(x|z) p(h(x)|x) \log \hat{p}(z | h(x)) dx \\
&= -\sum_{z=1}^{C} \sum_{h(x)=1}^{K} \int p(z) p(x|z) p(h(x)|x) \log \frac{\exp(f_{h(x)}^\top W_z)}{\sum_{i=1}^{C} \exp(f_{h(x)}^\top W_i)} dx \\
&= \sum_{z=1}^{C} p(z) \mathbb{E}_{x \sim \mathcal{P}_z} \left[\sum_{y=1}^{K} p(y|x) \big(-v_y^\top W_z + \log \sum_{i=1}^{C} \exp(v_y^\top W_i) \big) \right]
\end{aligned}$$

Furthermore, since the W learned over collapsed embeddings satisfies $W_z = v_y$ for $S(z) = y$, we have that $\log \sum_{i=1}^{C} \exp(v_y^\top W_i) = m \exp(1) + (C - m) \exp(c_K)$ for any y, and our expected

generalization error is

$$\sum_{z=1}^{C} p(z)\mathbb{E}_{x\sim\mathcal{P}_z}\left[-p(y=S(z)|x)-p(y\neq S(z)|x)\delta+\log(m\exp(1)+(C-m)\exp(c_K))\right]$$

$$=\log(m\exp(1)+(C-m)\exp(c_K))-c_K-(1-c_K)\sum_{z=1}^{C} p(z)\mathbb{E}_{x\sim\mathcal{P}_z}\left[p(y=S(z)|x)\right]$$

This tells us that the generalization error is at most $\log(m\exp(1)+(C-m)\exp(c_K))-c_K$ and at least $\log(m\exp(1)+(C-m)\exp(c_K))-1$.

For the original task, we can apply this same approach to the case where $m=1, C=K$ to get that the average generalization error is

$$\mathbb{E}_{h(x)}\left[\mathcal{L}(x,y,\hat{f}_1)\right] = \log(\exp(1)+(K-1)\exp(c_K)) - c_K-(1-c_K)\sum_{z=1}^{C} p(z)\mathbb{E}_{x\sim\mathcal{P}_z}\left[p(y=S(z)|x)\right]$$

This is at least $\log(\exp(1)+(K-1)\exp(c_K))-1$ and at most $\log(\exp(1)+(K-1)\exp(c_K))-c_K$. □

D PROOFS

D.1 PROOFS FOR SECTION 3.1

First, we characterize the optimal linear classifier (for both the coarse-to-fine transfer task and the original task) learned on the collapsed embeddings.

Lemma 4 (Downstream linear classifier for coarse-to-fine task)**.** *Suppose the dataset $\mathcal{D}_z$ is class-balanced across z, and the embeddings satisfy $f(x)=v_i$ if $h(x)=i$ where $\{v_i\}_{i=1}^K$ form the regular simplex. Then the optimal weight matrix $W^\star\in\mathbb{R}^{C\times d}$ that minimizes $\hat{\mathcal{L}}(W,\mathcal{D}_z)$ satisfies $W_z^\star=v_y$ for $y=S(z)$.*

Proof. Formally, the optimization problem we are solving is

$$\text{minimize } -\sum_{y=1}^{K}\sum_{z\in S_y}\log\frac{\exp(v_y^\top W_z)}{\sum_{j=1}^{C}\exp(v_y^\top W_j)} \tag{10}$$

$$\text{s.t. } \|W_z\|_2^2\leq 1\ \forall z\in\mathcal{Z} \tag{11}$$

The Lagrangian of this optimization problem is

$$\sum_{y=1}^{K}\sum_{z\in S_y}-v_y^\top W_z+m\sum_{y=1}^{K}\log\left(\sum_{j=1}^{C}\exp(v_y^\top W_j)\right)+\sum_{i=1}^{C}\lambda_i(\|W_i\|_2^2-1)$$

and the stationarity condition w.r.t. W_z is

$$-v_{S(z)}+m\sum_{y=1}^{K}\frac{v_y\exp(v_y^\top W_z)}{\sum_{j=1}^{C}\exp(v_y^\top W_j)}+2\lambda_z W_z=0 \tag{12}$$

Substituting $W_z=v_{S(z)}$, we get $-v_{S(z)}+m\sum_{y=1}^{K}\frac{v_y\exp(v_y^\top v_{S(z)})}{\sum_{j=1}^{C}\exp(v_y^\top v_{S(j)})}+2\lambda_z v_{S(z)}=0$. Using the fact that $v_i^\top v_j=\delta$ for all $i\neq j$, this equals $-v_{S(z)}+m\cdot\frac{v_{S(z)}\exp(1)+\exp(\delta)\sum_{y\neq S(z)}v_y}{m\exp(1)+(C-m)\exp(\delta)}+2\lambda_z v_{S(z)}=0$.

We next use the fact that $\sum_{i=1}^{K} v_i = 0$ to get that $\lambda_z = \frac{1}{2}\big(1 - m \cdot \frac{\exp(1)-\exp(\delta)}{m\exp(1)+(C-m)\exp(\delta)}\big) \geq 0$, satisfying the dual constraint. We can further verify complementary slackness and primal feasibility, since $\|W_z^\star\|_2^2 = 1$, to confirm that an optimal weight matrix satisfies $W_z^\star = v_y$ for $y = S(z)$.

□

Corollary 1. *When we apply the above proof to the case when $m = 1$, we recover that the optimal weight matrix $W^\star \in \mathbb{R}^{K\times d}$ that minimizes $\hat{\mathcal{L}}(W, \mathcal{D})$ for the original task on $(x, y) \sim \mathcal{P}$ satisfies $W_y^\star = v_y$ for all $y \in \mathcal{Y}$.*

We now prove Observation 1 and 2. Then, we present an additional result on transfer learning on collapsed embeddings to general tasks of the form $(x', y') \sim \mathcal{P}'$.

Observation 1. *Class collapse minimizes $\mathcal{L}(x, z, f)$ if for all x, 1) $p(y = h(x)|x) = 1$, meaning that each x is deterministically assigned to one class, and 2) $p(z|x) = \frac{1}{m}$ where $z \in S_{h(x)}$. The second condition implies that $p(x|z) = p(x|y)$ for all $z \in S_y$, meaning that there is no distinction among strata from the same class. This contradicts our generative model assumptions.*

Proof. We write out the generalization error for the downstream task, $\mathcal{L}(x, z, f) = \mathbb{E}_{x,z}\left[-\log \hat{p}(z|x)\right]$ using our conditions that $p(y = h(x)|x) = 1$ and $p(z|x) = \frac{1}{m}$.

$$\begin{aligned}
\mathcal{L}(x, z, f) &= -\int p(x) \sum_{z=1}^{C} p(z|x) \log \hat{p}(z|f(x)) dx \\
&= -\int p(x) \sum_{z=1}^{C} p(z|x) \log \frac{\exp(f(x)^\top W_z)}{\sum_{i=1}^{C} \exp(f(x)^\top W_i)} dx \\
&= -\sum_{y=1}^{K} \int_{x:h(x)=y} p(x) \cdot \frac{1}{m} \sum_{z\in S_y} \log \frac{\exp(f(x)^\top W_z)}{\sum_{i=1}^{C} \exp(f(x)^\top W_i)}
\end{aligned}$$

To minimize this, $f(x)$ should be the same across all x where $h(x)$ is the same value, since $p(z|x)$ does not change across fixed $h(x)$ and thus varying $f(x)$ will not further decrease the value of this expression. Therefore, we rewrite $f(x)$ as $f_{h(x)}$. Using the fact that y is class balanced, our loss is now

$$\begin{aligned}
\mathcal{L}(x, y, z) &= -\frac{1}{m} \sum_{y=1}^{K} \sum_{z\in S_y} \int_{x:h(x)=y} p(x) \log \frac{\exp(f_{h(x)}^\top W_z)}{\sum_{i=1}^{C} \exp(f_{h(x)}^\top W_z)} dx \\
&= -\frac{1}{C} \sum_{y=1}^{K} \sum_{z\in S_y} \log \frac{\exp(f_y^\top W_z)}{\sum_{i=1}^{C} \exp(f_y^\top W_i)}
\end{aligned}$$

We claim that $f_y = v_y$ and $W_z = v_y$ for all $S(z) = y$ minimizes this expression. The corresponding Lagrangian is

$$\sum_{y=1}^{K} \sum_{z\in S_y} -f_y^\top W_z + m \sum_{y=1}^{K} \log\left(\sum_{i=1}^{C} \exp(f_y^\top W_i)\right) + \sum_{y=1}^{K} \nu_y(\|f_y\|_2^2 - 1) + \sum_{i=1}^{C} \lambda_i(\|W_i\|_2^2 - 1)$$

The stationarity condition with respect to W_z is the same as (12), and we have already demonstrated that the feasibility constraints and complementary slackness are satisfied on W. The stationarity condition with respect to f_y is

$$-\sum_{z\in S_y} W_z + m \cdot \frac{\sum_{i=1}^{C} W_i \exp(f_y^\top W_i)}{\sum_{i=1}^{C} \exp(f_y^\top W_i)} + 2\lambda_y f_y = 0$$

Substituting in $W_i = v_{S(i)}$ and $f_y = v_y$, we get $-\sum_{z\in S_y} v_y + m \cdot \frac{\sum_{i=1}^{C} v_{S(i)} \exp(v_y^\top v_{S(i)})}{\sum_{i=1}^{C} \exp(v_y^\top v_{S(i)})} + 2\lambda_y v_y = 0$. Using the definition of the regular simplex, this simplifies to $-mv_y + m\frac{mv_y \exp(1) - mv_y \exp(\delta)}{m\exp(1)+(C-m)\exp(\delta)} +$

$2\lambda_y v_y = 0$. We thus have that $\lambda_y = \frac{m}{2}\left(1 - \frac{m(\exp(1)-\exp(\delta))}{m\exp(1)+(C-m)\exp(\delta)}\right)$, and the feasibility constraints are satisfied. Therefore, $f_y = W_z = v_y$ for $y = S(z)$ minimizes the generalization error $\mathcal{L}(x, z, f)$ when $p(h(x)|x) = 1$ and $p(z|x) = \frac{1}{m}$.

Because $p(z|x) = \frac{1}{m}$ and $p(y = h(x)|x) = 1$, this means that $p(z) = \int_{x:h(x)=S(z)} p(z, x)dx = \frac{1}{m}\int_{x:h(x)=S(z)} p(x) = \frac{1}{mK} = \frac{1}{C}$. $p(z)$ being class balanced means that $p(x|z) = \frac{p(z|x)p(x)}{p(z)} = Kp(x) = \frac{p(y|x)p(x)}{p(y)} = p(x|y)$. Therefore, this condition suggests that there is no distinction among the strata within a class.

□

Observation 2. *Class collapse minimizes $\mathcal{L}(x, y, f)$ if, for all x, $p(y = h(x)|x) = 1$. This contradicts our generative model assumptions.*

Proof. This observation follows directly from Observation 1 by repeating the proof approach with $z = y, m = 1$.

Lastly, suppose it is not true that $p(y = h(x)|x) = 1$. Then, the generalization error on the original task is $\mathcal{L}(x, y, f) = -\int_{\mathcal{X}} \sum_{y=1}^{K} p(x)p(y|x) \log \hat{p}(y|f(x))$, which is minimized when $\hat{p}(y|f(x)) = p(y|x)$. Intuitively, a model constructed with label information, $\hat{p}(y|h(x))$, will not improve over one that uses x itself to approximate $p(y|x)$.

□

D.2 Proof of Theorem 1

We first demonstrate that the limit of $L_{spread}(f, n^+, n^-)$ under Definition 2 has the decomposition presented in our theorem. In $L_{attract}$, we divide the numerator and denominator by n^-, and in L_{repel} we divide the numerator and denominator by n^+:

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

$$L_{repel}(f, n^+) = \mathbb{E}\left[-\log \frac{\exp(\sigma(x, x^{aug}))}{\frac{1}{n^+}\exp(\sigma(x, x^{aug})) + \frac{1}{n^+}\sum_{i=1}^{n^+}\exp(\sigma(x, x_i^+))}\right] + \log n^+$$

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

$$\begin{aligned} L_{spread}(f, n^+, n^-) - \alpha \log n^- - (1-\alpha)\log n^+ &= -\alpha\mathbb{E}\left[\sigma(x, x^+)\right] - (1-\alpha)\mathbb{E}\left[\sigma(x, x^{aug})\right] \\ &+ \alpha\mathbb{E}\left[\log\left(\frac{1}{n^-}\exp(\sigma(x, x^+)) + \frac{1}{n^-}\sum_{i=1}^{n^-}\exp(\sigma(x, x_i^-))\right)\right] \\ &+ (1-\alpha)\mathbb{E}\left[\log\left(\frac{1}{n^+}\exp(\sigma(x, x^{aug})) + \frac{1}{n^+}\sum_{i=1}^{n^+}\exp(\sigma(x, x_i^+))\right)\right] \end{aligned}$$

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

$$\begin{aligned} &\lim_{n^+, n^- \to \infty} L_{spread}(f, n^+, n^-) - \alpha \log n^- - (1-\alpha)\log n^+ \\ &= -\alpha\mathbb{E}\left[\sigma(x, x^+)\right] - (1-\alpha)\mathbb{E}\left[\sigma(x, x^{aug})\right] \\ &+ \alpha\mathbb{E}_x\left[\log \mathbb{E}_{x^-}\left[\exp(\sigma(x, x^-))\right]\right] + (1-\alpha)\mathbb{E}_x\left[\log \mathbb{E}_{x^+}\left[\exp(\sigma(x, x^+))\right]\right] \\ &= L_{align}(f) + L_{neg}(f) + L_{uniform}(f) \end{aligned}$$

Next, we analyze these individual loss components. Following Wang & Isola (2020)'s notation, define

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

$L_{align}(f)$ This loss component is $L_{align}(f) = -\big(\alpha\mathbb{E}\left[\sigma(x, x^+)\right] + (1-\alpha)\mathbb{E}\left[\sigma(x, x^{aug})\right]\big)$. This is minimized by making each $\sigma(x, x^+)$ and $\sigma(x, x^{aug})$ as large as possible, e.g. $f(x) = f(x^+) = f(x^{aug})$ for each anchor x.

$L_{uniform}(f)$ This loss component is $L_{uniform}(f) = (1-\alpha)\mathbb{E}_x\left[\log \mathbb{E}_{x^+}\left[\exp(\sigma(x, x^+))\right]\right]$. Conditioning on the label of x and using the definition of x^+, we can write this as

$$\begin{aligned} L_{uniform}(f) &= (1-\alpha)\mathbb{E}_y\left[\mathbb{E}_{x|h(x)=y}\left[\log \mathbb{E}_{x^+|h(x^+)=y}\left[\exp(\sigma(x, x^+))\right]\right]\right] \\ &= (1-\alpha)\sum_{i\in\mathcal{Y}} p(y=i)\int p(x|h(x)=i)\bigg(\log\int p(x^+|h(x^+)=i)\exp(\sigma(x, x^+))dx^+\bigg)dx \end{aligned}$$

To minimize this, we look at a relaxation of optimizing over a set of K measures, $\{\mu_i\}_{i=1}^K$ where each μ_i represents the class conditional distribution with density $p(x|h(x)=i)$ (since the encoder is assumed to be infinitely powerful). Then, to minimize $L_{uniform}(f)$, we want to solve

$$\begin{aligned} &\text{minimize}_{\{\mu_i\}_{i=1}^K} \sum_{i\in\mathcal{Y}} p(y=i)\int\bigg(\log\int\exp(\sigma(x, x^+))d\mu_y(x^+)\bigg)d\mu_i(x) \\ =&\text{minimize}_{\{\mu_i\}_{i=1}^K} \sum_{i\in\mathcal{Y}} p(i)\int \log U_{\mu_i}(u)d\mu_i(u) \end{aligned}$$

Since there are no pairwise terms where measures of different classes interact, the minimum of the above expression is obtained with $\mu_i^\star = \text{argmin}_{\mu_i}\int \log U_{\mu_i}(u)d\mu_i(u)$ for all $i \in \mathcal{Y}$. This is the exact form of the expression that Wang & Isola (2020) analyzes for $L_{uniform}$, where they use a measure μ on the entire dataset rather than a class-conditional μ_i. Therefore, using their analysis approach, we can directly conclude that $\mu_i^\star = \sigma_{d-1}$, the normalized surface area measure on $\mathbb{S}^{d-1}$. Therefore, this component is minimized when points of each class are distributed uniformly on the hypersphere.

$L_{neg}(f)$ This loss component is $L_{neg}(f) = \alpha\mathbb{E}_x\left[\log \mathbb{E}_{x^-}\left[\exp(\sigma(x, x^-))\right]\right]$. Conditioning on the label of x and using the definition of x^-, we can write this as

$$\begin{aligned} L_{neg}(f) &= \alpha\mathbb{E}_y\left[\mathbb{E}_{x|h(x)=y}\left[\log \mathbb{E}_{x^-|h(x^-)\neq y}\left[\exp(\sigma(x, x^-))\right]\right]\right] \\ &= \alpha\sum_{i\in\mathcal{Y}} p(y=i)\int p(x|y=i)\bigg(\log\int p(x^-|y\neq i)\exp(\sigma(x, x^-))dx^-\bigg)dx \end{aligned}$$

We consider a relaxation of this problem into K one-versus-all problems. That is, we compute optimal pairs of measures corresponding to $p(x|y=i)$, $p(x|y\neq i)$ for each i. Let y' be an indicator variable for if $y = i$ or not. For notation, define $\rho(0) = p(y'=0)$ and $\rho(1) = p(y'=1)$. Our problem is now equivalent to analyzing the binary setting with the following objective function to minimize:

$$\begin{aligned} &\rho(0)\int p(x|y'=0)\bigg(\log\int p(x^-|y'=1)\exp(\sigma(x, x^-))dx^-\bigg)dx \\ +&\rho(1)\int p(x|y'=1)\bigg(\log\int p(x^-|y'=0)\exp(\sigma(x, x^-))dx^-\bigg)dx \end{aligned}$$

Since the encoder is assumed to be infinitely powerful, we can consider optimizing over the class-conditional measures μ_0 and μ_1 in $\mathcal{M}(\mathbb{S}^{d-1})$, the set of Borel probability measures on $\mathbb{S}^{d-1}$. The optimization problem is now

$$\begin{aligned} \text{minimize}_{\mu_0,\mu_1}\quad &\rho(0)\int\bigg(\log\int\exp(\sigma(x, x^-))d\mu_1(x)\bigg)d\mu_0(x) \\ &+\int\bigg(\log\int\exp(\sigma(x, x^-))d\mu_0(x)\bigg)d\mu_1(x) \end{aligned}$$

The expression we want to minimize is thus

$$\text{minimize}_{\mu_i,\mu_{-i}} \rho(0) \int \log U_{\mu_1}(u) d\mu_0(u) + \rho(1) \int \log U_{\mu_0}(u) d\mu_1(u) \tag{13}$$

Following the approach of Wang & Isola (2020), we analyze the distributions $\mu_0^\star, \mu_1^\star$ that minimize this expression in three steps. First, we show that the minimum of (13) exists, i.e. the infimum is attained for some two measures. Second, we show that $U_{\mu_0^\star}$ is constant $\mu_1^\star$-almost surely, and vice versa. Lastly, we use this to show that the collapsed embeddings distribution minimizes (13).

1. **Minimizers of** (13) **exist.**
   Let m be a sequence such that

$$\begin{aligned}&\lim_{m\to\infty} \rho(0) \int \log U_{\mu_1^m}(u) d\mu_0^m(u) + \rho(1) \int \log U_{\mu_0^m}(u) d\mu_1^m(u) \\ &= \inf_{\mu_0,\mu_1} \rho(0) \int \log U_{\mu_1}(u) d\mu_0(u) + \rho(1) \int \log U_{\mu_0}(u) d\mu_1(u)\end{aligned}$$

   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 poinnt $(\mu_0^\star, \mu_1^\star)$. Because $\{\log U_{\mu_0^n}\}_n$ is uniformly bounded and continuously convergent to $\log U_{\mu_0^\star}$ and same for μ_1^n and $\mu_1^\star$, it holds that

$$\begin{aligned}&\rho(0) \int \log U_{\mu_1^\star}(u) d\mu_0^\star(u) + \rho(1) \int \log U_{\mu_0^\star}(u) d\mu_1^\star(u) \\ &= \lim_{n\to\infty} \rho(0) \int \log U_{\mu_1^n}(u) d\mu_0^n(u) + \rho(1) \int \log U_{\mu_0^n}(u) d\mu_1^n(u))\end{aligned}$$

   and therefore $\mu_0^\star, \mu_1^\star$ achieve the infimum of (13).
2. $U_{\mu_1^\star}$ **is constant** $\mu_0^\star$**-almost surely and** $U_{\mu_0^\star}$ **is constant** $\mu_1^\star$**-almost surely, for any minimizer** $(\mu_0^\star, \mu_1^\star)$ **of** (13)**.**
   Formally, define $(\mu_0^\star, \mu_1^\star)$ to be a solution of (13), i.e.

$$\mu_0^\star, \mu_1^\star \in \text{argmin}_{\mu_0,\mu_1} \rho(0) \int \log U_{\mu_1}(u) d\mu_0(u) + \rho(1) \int \log U_{\mu_0}(u) d\mu_1(u)$$

   Define the Borel sets where $\mu_i^\star$ has positive measure to be $\mathcal{T}_i = \{T \in \mathcal{M}(\mathbb{S}^{d-1}) : \mu_i^\star(T) > 0\}$. Define the conditional distribution of $\mu_i^\star$ on T for some $T \in \mathcal{T}_i$ as $\mu_{i,T}^\star$, where $\mu_{i,T}^\star(A) = \frac{\mu_i^\star(A \cap T)}{\mu_i^\star(T)}$.
   Now we consider a mixture $(1-\alpha)\mu_0^\star + \alpha\mu_{0,T}^\star$. The first variation of $\mu_0^\star$ states that

$$\begin{aligned}0=&\frac{\partial}{\partial\alpha}\left[\rho(0) \int \log U_{\mu_1^\star}(u) d((1-\alpha)\mu_0^\star + \alpha\mu_{0,T}^\star)(u) + \rho(1) \int \log U_{(1-\alpha)\mu_0^\star+\alpha\mu_{0,T}^\star} d\mu_1^\star(u)\right|_{\alpha=0}\Bigg] \\ &= \rho(0) \int \log U_{\mu_1^\star}(u) d(\mu_{0,T}^\star - \mu_0^\star)(u) + \rho(1) \int \frac{U_{\mu_{0,T}^\star}(u) - U_{\mu_0^\star}(u)}{U_{\mu_0^\star}(u)} d\mu_1^\star(u),\end{aligned}$$

   Where we've used the fact that $\frac{\partial}{\partial\alpha} U_{(1-\alpha)\mu_0^\star+\alpha\mu_{0,T}^\star}(u)\big|_{\alpha=0} = \frac{\partial}{\partial\alpha} \int \exp(u^\top v/\tau) d((1-\alpha)\mu_0^\star + \alpha\mu_{0,T}^\star)(v)\big|_{\alpha=0} = U_{\mu_{0,T}^\star}(u) - U_{\mu_0^\star}(u)$. Therefore, due to symmetry the optimality conditions using the first variation are

$$\rho(0) \int \log U_{\mu_1^\star}(u) d(\mu_{0,T}^\star - \mu_0^\star)(u) + \rho(1) \int \frac{U_{\mu_{0,T}^\star}(u)}{U_{\mu_0^\star}(u)} d\mu_1^\star(u) = \rho(1) \tag{14}$$

$$\rho(1) \int \log U_{\mu_1^\star}(u) d(\mu_{1,T}^\star - \mu_1^\star)(u) + \rho(0) \int \frac{U_{\mu_{1,T}^\star}(u)}{U_{\mu_1^\star}(u)} d\mu_0^\star(u) = \rho(0) \tag{15}$$

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

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

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

$$\lim_{n\to\infty}\int U_{\mu_0^\star}(u)d\mu^\star_{1,T_1^n}(u) = \sup_{T_1\in\mathcal{T}_1}\int U_{\mu_0^\star}(u)d\mu^\star_{1,T_1}(u) = U^\star_{0,1}$$

It holds that $\mu_0^\star(\{u : U_{\mu_1^\star}(u) \geq U^\star_{1,0}\}) = 0$, $\mu^\star_{0,T_0^n}(\{u : U_{\mu_1^\star}(u) \geq U^\star_{1,0}\}) = 0$ and similarly $\mu_1^\star(\{u : U_{\mu_0^\star}(u) \geq U^\star_{0,1}\}) = 0$, $\mu^\star_{1,T_1^n}(\{u : U_{\mu_0^\star}(u) \geq U^\star_{0,1}\}) = 0$.
This implies that asymptotically $U_{\mu_0^\star}$ is constant $\mu^\star_{1,T_1^n}$-almost surely:

$$\begin{aligned}
&\int \left| U_{\mu_0^\star}(u) - \int U_{\mu_0^\star}(u')d\mu_{1,T_1^n}(u')\right| d\mu^\star_{1,T_1^n}(u)\\
=&2\int \max\left(0, U_{\mu_0^\star}(u) - \int U_{\mu_0^\star}(u')d\mu^\star_{1,T_1^n}(u')\right)d\mu^\star_{1,T_1^n}(u)\\
\leq&2\left(U^\star_{0,1} - \int U_{\mu_0^\star}(u)d\mu^\star_{1,T_1^n}(u)\right) \to 0
\end{aligned}$$

And the same holds that $U_{\mu_1^\star}$ is constant $\mu^\star_{0,T_0^n}$-almost surely. As a result, $\lim_{n\to\infty}\int \log U_{\mu_0^\star}(u)d\mu^\star_{1,T_1^n}(u) = \log U^\star_{0,1}$ and $\lim_{n\to\infty}\int \log U_{\mu_1^\star}(u)d\mu^\star_{0,T_0^n}(u) = \log U^\star_{1,0}$.
We now revisit (14) with a mixture over $\mu_0^\star$ and μ_{0,T_0^n}:

$$\begin{aligned}
\rho(1) &= \rho(0)\int \log U_{\mu_1^\star}(u)d(\mu^\star_{0,T_0^n} - \mu_0^\star)(u) + \rho(1)\int \frac{U_{\mu^\star_{0,T_0^n}}(u)}{U_{\mu_0^\star}(u)}d\mu_1^\star(u)\\
&\geq \rho(0)\int \log U_{\mu_1^\star}(u)d(\mu^\star_{0,T_0^n} - \mu_0^\star)(u) + \frac{\rho(1)}{U^\star_{0,1}}\int U_{\mu_1^\star}(u)d\mu^\star_{0,T_0^n}(u)
\end{aligned}$$

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

$$\rho(1) \geq \rho(0)\log U^\star_{1,0} - \rho(0)\int \log U_{\mu_1^\star}(u)d\mu_0^\star(u) + \frac{\rho(1)}{U^\star_{0,1}}U^\star_{1,0}$$

and rearranging and doing the same to (15) yields

$$\begin{aligned}
\frac{\rho(1)}{\rho(0)}\Big(1 - \frac{U^\star_{1,0}}{U^\star_{0,1}}\Big) &\geq \log U^\star_{1,0} - \int \log U_{\mu_1^\star}(u)d\mu_0^\star(u)\\
\frac{\rho(0)}{\rho(1)}\Big(1 - \frac{U^\star_{0,1}}{U^\star_{1,0}}\Big) &\geq \log U^\star_{0,1} - \int \log U_{\mu_0^\star}(u)d\mu_1^\star(u)
\end{aligned}$$

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

$$\begin{aligned}
\int \log U_{\mu_1^\star}(u)d\mu_0^\star(u) &= \log\int U_{\mu_1^\star}(u)d\mu_0^\star(u)\\
\int \log U_{\mu_0^\star}(u)d\mu_1^\star(u) &= \log\int U_{\mu_0^\star}(u)d\mu_1^\star(u)
\end{aligned}$$

3. **Collapsed embeddings minimize L_{neg}.**
We return to the original objective function (13). Now, the optimal value of this expression can be written as

$$\begin{aligned}
&\rho(0)\log\int U_{\mu_1^\star}(u)d\mu_0^\star(u) + \rho(1)\log\int U_{\mu_0^\star}(u)d\mu_1^\star(u)\\
=&\log\int U_{\mu_1^\star}(u)d\mu_0^\star(u) = \log\int\int \exp(u^\top v/\tau)d\mu_0^\star(u)d\mu_1^\star(v)
\end{aligned}$$

We can see that when $K = 2$ our optimal distribution is to have collapsed embeddings with dot product -1, i.e. opposite of each other on the hypersphere. For larger K, μ_0 and μ_1 represent a one-versus-all scenario. Collectively, having embeddings per class with probability 1 on a single point is part of the optimal solution for one-versus-all measures since there is nothing in these objective functions that enforce intra-class spread. We abuse notation and refer to μ_i as the embedding for class i. Given this property, we can revisit L_{neg}; minimizing it is equivalent to minimizing

$$\sum_{i=1}^{K} \log \sum_{j \neq k} \exp(\mu_i^\top \mu_j / \tau)$$

We can show using KKT conditions that setting $\mu_i = v_i$ for all i is an optimal distribution. In particular, the Lagrangian is $\sum_{i=1}^{K} \log \sum_{j \neq i} \exp(\mu_i^\top \mu_j / \tau) + \sum_{i=1}^{K} \lambda_i (\|\mu_i\|^2 - 1)$, and the stationarity condition is

$$\frac{\sum_{j \neq k} \frac{\mu_j}{\tau} \exp(\mu_k^\top \mu_j / \tau)}{\sum_{j \neq k} \exp(\mu_k^\top \mu_j / \tau)} + \sum_{i \neq k} \frac{\frac{\mu_k}{\tau} \exp(\mu_i^\top \mu_k / \tau)}{\sum_{j \neq i} \exp(\mu_i^\top \mu_j / \tau)} + 2\lambda_k \mu_k = 0 \; \forall k$$

This has a solution of $\lambda_i = \frac{\frac{1}{\tau} \exp(\delta/\tau)}{(K-1)\exp(\delta/\tau)}$, and all other conditions hold. Therefore, we have shown that collapsed embeddings minimize L_{neg}.

D.3 Proofs for Section 4.2

Lemma 1. *Denote $\hat{f}_1$ to be an encoder learned on $\mathcal{D}$ with N using L_{spread}. Using $\hat{f}_1$ and a mean classifier where $W_z = \mathbb{E}_{x \sim \mathcal{P}_z}[\hat{f}_1(x)]$, the generalization error on the coarse-to-fine transfer task is at most*

$$\mathcal{L}(x, z, \hat{f}_1) \leq \mathbb{E}_z \left[\log \Big(\sum_{i:S(i)=S(z)} \exp(-\delta_{intra}(z,i)) + \sum_{i:S(i) \neq S(z)} \exp(-\delta_{inter}(z,i)) \Big) \right] - 1$$

where $\delta_{intra}(z,i) = \lambda\Big(\frac{1}{2}\delta(\hat{f}_1, z, i)^2 - 1\Big)$ and $\delta_{inter}(z,i) = \lambda\Big(\frac{1}{2}\|\mathbb{E}_z[\hat{f}_1(x)] - \mathbb{E}_i[\hat{f}_1(x)]\|^2 - 1\Big)$ for some $\lambda > 0$.

Proof. The generalization error is

$$\begin{aligned}
\mathcal{L}(x, z, \hat{f}_1) &= -\mathbb{E}_z \left[\mathbb{E}_{x \sim \mathcal{P}_z} \left[\log \frac{\exp(\hat{f}_1(x)^\top W_z)}{\sum_{i=1}^{C} \exp(\hat{f}_1(x)^\top W_i)} \right] \right] \\
&= \mathbb{E}_z \left[\mathbb{E}_{x \sim \mathcal{P}_z} \left[-\hat{f}_1(x)^\top W_z + \log \sum_{i=1}^{C} \exp(\hat{f}_1(x)^\top W_i) \right] \right]
\end{aligned}$$

Using the definition of the mean classifier,

$$\begin{aligned}
\mathcal{L}(x, z, \hat{f}_1) &= \mathbb{E}_z \left[-1 + \mathbb{E}_{x \sim \mathcal{P}_z} \left[\log \sum_{i=1}^{C} \exp\left(\hat{f}_1(x)^\top \mathbb{E}_{x \sim \mathcal{P}_i}[\hat{f}_1(x)] \right) \right] \right] \\
&= -1 + \mathbb{E}_z \left[\mathbb{E}_{x \sim \mathcal{P}_z} \left[\log \sum_{i=1}^{C} \exp\left(\hat{f}_1(x)^\top \mathbb{E}_i[\hat{f}_1(x)] \right) \right] \right]
\end{aligned}$$

Since $\hat{f}_1(x)$ is bounded, there exists a constant $\lambda > 0$ such that $\mathbb{E}_{x \sim \mathcal{P}_z} \Big[\log \sum_{i=1}^{C} \exp\left(\hat{f}_1(x)^\top \mathbb{E}_i[\hat{f}_1(x)] \right) \Big] \leq \log \Big(\sum_{i=1}^{C} \exp\left(\lambda \mathbb{E}_z[\hat{f}_1(x)]^\top \mathbb{E}_i[\hat{f}_1(x)] \right) \Big)$. We

can also rewrite the dot product between mean embeddings per strata in terms of the distance between them:

$$\mathcal{L}(x, z, \hat{f}_1) \leq -1 + \mathbb{E}_z \left[\log \Big(\sum_{i=1}^{C} \exp \left(\lambda \mathbb{E}_z[\hat{f}_1(x)]^\top \mathbb{E}_i[\hat{f}_1(x)] \right) \Big) \right]$$
$$= -1 + \mathbb{E}_z \left[\log \Big(\sum_{i=1}^{C} \exp \Big(-\frac{\lambda}{2} \|\mathbb{E}_z[\hat{f}_1(x)] - \mathbb{E}_i[\hat{f}_1(x)]\|^2 + \lambda \Big) \Big) \right]$$

Note that when i and z satisfy $S(i) = S(z)$, $\|\mathbb{E}_z[\hat{f}_1(x)] - \mathbb{E}_i[\hat{f}_1(x)]\|$ can be written as $\delta(\hat{f}_1, z, i)$, which we have analyzed. Therefore,

$$\mathcal{L}(x, z, \hat{f}_1) \leq -1 + \mathbb{E}_z \Bigg[\log \Big(\sum_{i \in S_{S(z)}} \exp \Big(-\frac{\lambda}{2} \delta(\hat{f}_1, z, i)^2 + \lambda \Big)$$
$$+ \sum_{i : S(i) \neq S(z)} \exp \Big(-\frac{\lambda}{2} \|\mathbb{E}_z[\hat{f}_1(x)] - \mathbb{E}_i[\hat{f}_1(x)]\|^2 + \lambda \Big) \Big) \Bigg]$$

This directly gives us our desired bound.

□

Lemma 2. *Denote $\hat{f}_1$ to be an encoder learned on $\mathcal{D}$ with N using L_{spread}. Using $\hat{f}_1$ and a mean classifier where $W_y = \mathbb{E}_{x \sim p(\cdot|y)}[\hat{f}_1(x)]$, the generalization error on the original task is at most*

$$\mathcal{L}(x, y, \hat{f}_1) \leq \mathbb{E}_z \left[\log \Big(\exp(-\delta_{intra}(z)) + \sum_{i \neq S(z)} \exp(-\delta_{inter}(z)) \Big) + \frac{1}{\lambda_y} \delta_{intra}(z) \right]$$

where $\delta_{intra}(z) = \eta \sum_{z' \in S_{S(z)}} p(z'|S(z)) \Big(\frac{1}{2} \delta(\hat{f}_1, z, z')^2 - 1 \Big)$ *and* $\delta_{inter}(z) = \eta \sum_{z' \notin S(z)} p(z'|S(z')) \big(\frac{1}{2} \|\mathbb{E}_z[\hat{f}_1(x)] - \mathbb{E}_{z'}[\hat{f}_1(x)]\|^2 - 1$ *for some $\eta > 0$.*

Proof. The generalization error is

$$\mathcal{L}(x, y, \hat{f}_1) = -\mathbb{E}_z \left[\mathbb{E}_{x \sim \mathcal{P}_z} \left[\log \frac{\exp(\hat{f}_1(x)^\top W_{S(z)})}{\sum_{i=1}^{K} \exp(\hat{f}_1(x)^\top W_i)} \right] \right]$$
$$= \mathbb{E}_z \left[\mathbb{E}_{x \sim \mathcal{P}_z} \left[-\hat{f}_1(x)^\top W_{S(z)} + \log \sum_{i=1}^{K} \exp(\hat{f}_1(x)^\top W_i) \right] \right]$$

We substitute in the definition of the mean classifier to get

$$\mathcal{L}(x, y, \hat{f}_1) = \mathbb{E}_z \Bigg[-\sum_{z' \in S_{S(z)}} p(z'|S(z)) \mathbb{E}_z[\hat{f}_1(x)]^\top \mathbb{E}_{z'}[\hat{f}_1(x)]$$
$$+ \mathbb{E}_{x \sim \mathcal{P}_z} \left[\log \sum_{i=1}^{K} \exp \Big(\sum_{z' \in S_i} p(z'|S_i) \hat{f}_1(x)^\top \mathbb{E}_{z'}[\hat{f}_1(x)] \Big) \right] \Bigg]$$

We can rewrite the dot product between mean embeddings per strata in terms of the distance between them:

$$\mathcal{L}(x, y, \hat{f}_1) = \mathbb{E}_z \Bigg[\sum_{z' \in S_{S(z)}} p(z'|S(z)) \cdot \Big(\frac{1}{2} \|\mathbb{E}_z[\hat{f}_1(x)] - \mathbb{E}_{z'}[\hat{f}_1(x)]\|^2 - 1 \Big)$$
$$+ \mathbb{E}_{x \sim \mathcal{P}_z} \left[\log \sum_{i=1}^{K} \exp \Big(\sum_{z' \in S_i} p(z'|S_i) \hat{f}_1(x)^\top \mathbb{E}_{z'}[\hat{f}_1(x)] \Big) \right] \Bigg]$$

We can write $\|\mathbb{E}_z[\hat{f}_1(x)] - \mathbb{E}_{z'}[\hat{f}_1(x)]\|$ in the above expression as $\delta(\hat{f}_1, z, z')$, which we have analyzed:

$$\begin{aligned}\mathcal{L}(x, y, \hat{f}_1) =& \mathbb{E}_z\Big[\sum_{z' \in S_{S(z)}} p(z'|S(z)) \cdot \Big(\frac{1}{2}\delta(\hat{f}_1, z, z')^2 - 1\Big) \\ &+ \mathbb{E}_{x \sim \mathcal{P}_z}\Big[\log \sum_{i=1}^{K} \exp\Big(\sum_{z' \in S_i} p(z'|S_i)\hat{f}_1(x)^\top \mathbb{E}_{z'}[\hat{f}_1(x)]\Big)\Big]\Big]\end{aligned}$$

From our previous proof, there exists $\lambda > 0$ such that this is at most

$$\begin{aligned}\mathcal{L}(x, y, \hat{f}_1) \leq& \mathbb{E}_z\Big[\sum_{z' \in S_{S(z)}} p(z'|S(z)) \cdot \Big(\frac{1}{2}\delta(\hat{f}_1, z, z')^2 - 1\Big) \\ &+ \log\Big(\sum_{i=1}^{K} \exp(\sum_{z' \in S_i} p(z'|S_i)\lambda \mathbb{E}_z[\hat{f}_1(x)]^\top \mathbb{E}_{z'}[\hat{f}_1(x)])\Big)\Big] \\ =& \mathbb{E}_z\Big[\sum_{z' \in S_{S(z)}} p(z'|S(z)) \cdot \Big(\frac{1}{2}\delta(\hat{f}_1, z, z')^2 - 1\Big) \\ &+ \log\Big(\sum_{i=1}^{K} \exp\Big(\sum_{z' \in S_i} p(z'|S_i)\Big(-\frac{\lambda}{2}\|\mathbb{E}_z[\hat{f}_1(x)] - \mathbb{E}_{z'}[\hat{f}_1(x)]\|^2 + \lambda\Big)\Big)\Big)\Big]\end{aligned}$$

We can simplify the log term in this expression by considering if $i = S(z)$, in which case the distances between strata centers can be written using $\delta(\hat{f}_1, z, z')$:

$$\begin{aligned}&\log\Big(\sum_{i=1}^{K} \exp\Big(\sum_{z' \in S_i} p(z'|S_i)\Big(-\frac{\lambda}{2}\|\mathbb{E}_z[\hat{f}_1(x)] - \mathbb{E}_{z'}[\hat{f}_1(x)]\|^2 + \lambda\Big)\Big)\Big) \\ =& \log\Big(\exp\Big(\sum_{z' \in S_{S(z)}} p(z'|S(z))\Big(-\frac{\lambda}{2}\delta(\hat{f}_1, z, z')^2 + \lambda\Big)\Big) \\ &+ \sum_{i \neq S(z)} \exp\Big(\sum_{z' \in S_i}\Big(-\frac{\lambda}{2}\|\mathbb{E}_z[\hat{f}_1(x)] - \mathbb{E}_{z'}[\hat{f}_1(x)]\|^2 + \lambda\Big)\Big)\Big)\Big)\end{aligned}$$

Putting everything together, the generalization error is at most

$$\begin{aligned}\mathcal{L}(x, y, \hat{f}_1) \leq& \mathbb{E}_z\Big[\sum_{z' \in S_{S(z)}} p(z'|S(z)) \cdot \Big(\frac{1}{2}\delta(\hat{f}_1, z, z')^2 - 1\Big) \\ &+ \log\Big(\exp\Big(\sum_{z' \in S_{S(z)}} p(z'|S(z))\Big(-\frac{\lambda}{2}\delta(\hat{f}_1, z, z')^2 + \lambda\Big)\Big) \\ &+ \sum_{i \neq S(z)} \exp\Big(\sum_{z' \in S_i}\Big(-\frac{\lambda}{2}\|\mathbb{E}_z[\hat{f}_1(x)] - \mathbb{E}_{z'}[\hat{f}_1(x)]\|^2 + \lambda\Big)\Big)\Big)\Big)\end{aligned}$$

This directly gives us our desired bound.

□

E Additional Experimental Details

E.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.
- **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., 2010), 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.

E.2 HYPERPARAMETERS

For all model quality experiments for L_{spread}, we first fixed $\tau = 0.5$ and swept $\alpha \in [0.16, 0.25, 0.33, 0.5, 0.67]$. We then took the two best-performing values and swept $\tau \in [0.1, 0.3, 0.5, 0.7, 0.9]$. For L_{SC} and L_{SS}, we swept $\tau \in [0.1, 0.3, 0.5, 0.7, 0.9]$. Final hyperparameter values for (τ, α) for L_{spread} were $(0.9, 0.67)$ for CIFAR10, $(0.5, 0.16)$ for CIFAR10-coarse, $(0.5, 0.33)$ for CIFAR100, $(0.5, 0.25)$ for CIFAR100-Coarse, $(0.5, 0.25)$ for CIFAR100-Coarse-U, $(0.5, 0.5)$ for MNIST, $(0.5, 0.5)$ for MNIST-coarse, $(0.5, 0.5)$ for ISIC, and $(0.5, 0.5)$ for waterbirds.

For coarse-to-fine transfer learning, we fixed $\tau = 0.5$ for all losses and swept $\alpha \in [0.16, 0.25, 0.33, 0.5, 0.67]$. Final hyperparameter values for α were 0.25 for CIFAR10-Coarse, 0.25 for CIFAR100-Coarse, 0.25 for CIFAR100-Coarse-U, and 0.5 for MNIST-Coarse.

E.3 APPLICATIONS

We describe additional experimental details for the applications.

Robustness Against Worst-Group Performance We follow the evaluation of Sohoni et al. (2020). First, we train a model on the standard class labels. We evaluate different loss functions for this step, including L_{spread}, L_{SC}, and the cross entropy loss L_{CE}. Then we project embeddings of the training set using a UMAP projection (McInnes et al., 2018), and cluster points to discover unlabeled subgroups. Finally, we use the unlabeled subgroups in a Group-DRO algorithm to optimize worst-group robustness (Sagawa et al., 2019).

Robustness Against Noise We use the same training setup as we use to evaluate model quality, and introduce symmetric noise into the labels for the contrastive loss head. We train the cross entropy head with a fraction of the full training set. In Section 5.2, we report results from training with 20% labels to cross entropy. We report additional levels in Appendix F.

We detect noisy labels with a simple geometric heuristic: for each point, we compute the cosine similarity between the embedding of the point and the center of all the other points in the batch that have the same class. We compare this similarity value to the average cosine similarity with points in the batch from every other class, and rank the points by the difference between these two values. Points with incorrect labels have a small difference between these two values (they appear to be small strata, so they are far away from points of the same class). Given the noise level ϵ as an input, we

Table 2: Performance of L_{spread} compared to L_{SC} and using $L_{attract}$ on its own.

	End Model Perf.			
Dataset	L_{SS}	L_{SC}	$L_{attract}$	L_{spread}
CIFAR10	89.7	90.9	91.3	**91.5**
CIFAR100	68.0	67.5	68.9	**69.1**

rank the points by this heuristic and mark the ϵ fraction of the batch with the smallest scores as noisy. We then correct their labels by adopting the label of the closest cluster center.

Minimal Coreset Construction We use the publicly-available evaluation framework for coresets from Toneva et al. (2019).[1] We use the official repository from Paul et al. (2021)[2] to recreate their coreset algorithms.

Our coreset algorithm proceeds in two parts. First, we give each point a difficulty rating based on how likely we are to classify it correctly under partial training. Then we subsample the easiest points to construct minimal coresets.

First, we mirror the set up from our thought experiment and train with L_{spread} on random samples of $t\%$ of the CIFAR10 training set, taking three random samples for each of $t \in [10, 20, 50]$ (and we train the cross entropy head with 1% labeled data). For each run, we record which points are classified correctly by the cross entropy head at the end of training, and bucket points the training set by how often the point was correctly classified. To construct a coreset of size $t\%$, we iteratively remove points from the largest bucket in each class. Our strategy removes easy examples first from the largest coresets, but maintains a set of easy examples in the smallest coresets.

F Additional Experimental Results

In this section, we report three sets of additional experimental results: the performance of using $L_{attract}$ on its own to train models, sample complexity of L_{spread} compared to L_{SC}, and additional noisy label results (including a bonus de-noising algorithm).

F.1 Performance of $L_{attract}$

In an early iteration of this project, we experienced success with using $L_{attract}$ on its own to train models, before realizing the benefits of adding in an additional term to prevent class collapse. As an ablation, we report on the performance of using $L_{attract}$ on its own in Table 2. $L_{attract}$ can outperform L_{SC}, but L_{spread} outperforms both. We do not report the results here, but $L_{attract}$ also performs significantly worse than L_{SC} on downstream applications, since it more direclty encourages class collapse.

F.2 Sample Complexity

Figure 6 shows the performance of training ViT models with various amounts of labeled data for L_{spread}, L_{SC}, and L_{SS}. In these experiments, we train the cross entropy head with 1% labeled data to isolate the effect of training data on the contrastive losses themselves.

L_{spread} outperforms L_{SC} and L_{SS} throughout. At 10% labeled data, L_{spread} outperforms L_{SS} by 13.9 points, and outperforms L_{SC} by 0.5 points. By 100% labeled data (for the contrastive head), L_{spread} outperforms L_{SS} by 25.4 points, and outperforms L_{SC} by 10.3 points.

[1] https://github.com/mtoneva/example_forgetting

[2] https://github.com/mansheej/data_diet

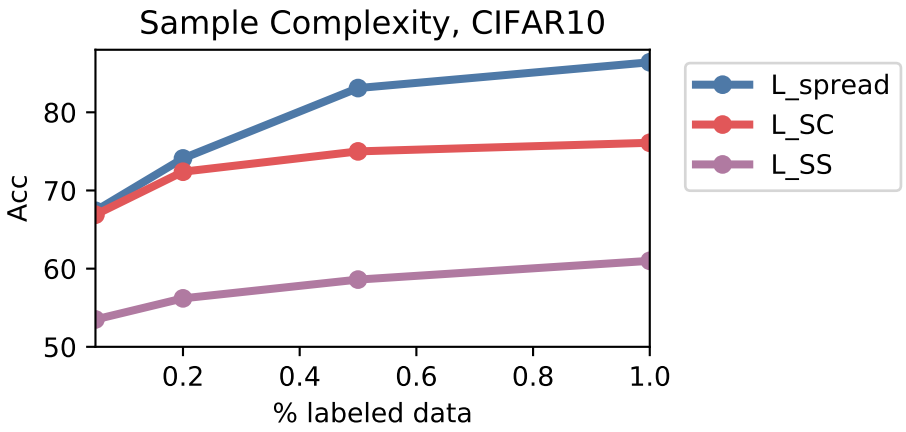

Figure 6: Performance of training ViT with L_{spread} compared to training with L_{SC} and L_{SS} on CIFAR10 at various amounts of labeled data. L_{spread} outperforms the baselines at each point. The cross entropy head here is trained with 1% labeled data to isolate the effect of training data on the contrastive losses.

F.3 Noisy Labels

In Section 5.2, we reported results from training the contrastive loss head with noisy labels and the cross entropy loss with clean labels from 20% of the training data.

In this section, we first discuss a de-noising algorithm inspired by Chuang et al. (2020) that we initially developed to correct for noisy labels, but that we did not observe strong empirical results from. We hope that reporting this result inspires future work into improving contrastive learning.

We then report additional results with larger amounts of training data for the cross entropy head.

F.3.1 Debiasing Noisy Contrastive Loss

First, we consider the triplet loss and show how to debias it in expectation under noise. Then we present an extension to supervised contrastive loss.

Noise-Aware Triplet Loss Consider the triplet loss:

$$L_{triplet} = \mathbb{E}_{\substack{x\sim\mathcal{P}, x^+\sim p^+(\cdot|x), \\ x^-\sim p^-(\cdot|x)}} \left[-\log \frac{\exp(\sigma(x, x^+))}{\exp(\sigma(x, x^+)) + \exp(\sigma(x, x^-))} \right] \tag{16}$$

Now suppose that we do not have access to true labels but instead have noisy labels denoted by the weak classifier $\widetilde{y} := \widetilde{h}(x)$. We adopt a simple model of symmetric noise where $\widetilde{p} = \Pr(\text{noisy label is correct})$.

We use $\widetilde{y}$ to construct $\widetilde{\mathcal{P}}^+$ and $\widetilde{\mathcal{P}}^-$ as $p(x^+|\widetilde{h}(x) = \widetilde{h}(x^+))$ and $p(x^-|\widetilde{h}(x) \neq \widetilde{h}(x^-))$. For simplicity, we start by looking at how the triplet loss in (16) is impacted when *noise is not addressed* in the binary setting. Define $L^{triplet}_{noisy}$ as $L_{triplet}$ used with $\widetilde{\mathcal{P}}^+$ and $\widetilde{\mathcal{P}}^-$.

Lemma 5. *When class-conditional noise is uncorrected, $L_{triplet}^{noisy}$ is equivalent to*

$$(\widetilde{p}^3 + (1-\widetilde{p})^3)L_{triplet} + \widetilde{p}(1-\widetilde{p})\mathbb{E}_{\substack{x\sim\mathcal{P} \\ x_1^+, x_2^+ \sim p^+(\cdot|x)}}\left[-\log\frac{\exp(\sigma(x,x_1^+))}{\exp(\sigma(x,x_1^+))+\exp(\sigma(x,x_2^+))}\right]$$
$$+ \widetilde{p}(1-\widetilde{p})\mathbb{E}_{\substack{x\sim\mathcal{P} \\ x_1^-, x_2^- \sim p^-(\cdot|x)}}\left[-\log\frac{\exp(\sigma(x,x_1^-))}{\exp(\sigma(x,x_1^-))+\exp(\sigma(x,x_2^-))}\right]$$
$$+ \widetilde{p}(1-\widetilde{p})\mathbb{E}_{\substack{x\sim\mathcal{P} \\ x^+\sim p^+(\cdot|x) \\ x^-\sim p^-(\cdot|x)}}\left[-\log\frac{\exp(\sigma(x,x^-))}{\exp(\sigma(x,x^+))+\exp(\sigma(x,x^-))}\right]$$

Proof. We split $L_{triplet}^{noisy}$ depending on if the noisy positive and negative pairs are truly positive and negative.

$$L_{triplet}^{noisy} = \mathbb{E}_{\substack{x\sim\mathcal{P} \\ \widetilde{x}^+\sim\widetilde{p}^+(\cdot|x) \\ \widetilde{x}^-\sim\widetilde{p}^-(\cdot|x)}}\left[-\log\frac{\exp(\sigma(x,\widetilde{x}^+))}{\exp(\sigma(x,\widetilde{x}^+))+\exp(\sigma(x,\widetilde{x}^-))}\right]$$
$$= p(h(x)=h(\widetilde{x}^+), h(x)\neq h(\widetilde{x}^-))\mathbb{E}_{\substack{x\sim\mathcal{P} \\ x^+\sim p^+(\cdot|x) \\ x^-\sim p^-(\cdot|x)}}\left[-\log\frac{\exp(\sigma(x,x^+))}{\exp(\sigma(x,x^+))+\exp(\sigma(x,x^-))}\right]$$
$$+ p(h(x)=h(\widetilde{x}^+), h(x)=h(\widetilde{x}^-))\mathbb{E}_{\substack{x\sim\mathcal{P} \\ x_1^+, x_2^+ \sim p^+(\cdot|x)}}\left[-\log\frac{\exp(\sigma(x,x_1^+))}{\exp(\sigma(x,x_1^+))+\exp(\sigma(x,x_2^+))}\right]$$
$$+ p(h(x)\neq h(\widetilde{x}^+), h(x)\neq h(\widetilde{x}^-))\mathbb{E}_{\substack{x\sim\mathcal{P} \\ x_1^-, x_2^- \sim p^-(\cdot|x)}}\left[-\log\frac{\exp(\sigma(x,x_1^-))}{\exp(\sigma(x,x_1^-))+\exp(\sigma(x,x_2^-))}\right]$$
$$+ p(h(x)\neq h(\widetilde{x}^+), h(x)=h(\widetilde{x}^-))\mathbb{E}_{\substack{x\sim\mathcal{P} \\ x^+\sim p^+(\cdot|x) \\ x^-\sim p^-(\cdot|x)}}\left[-\log\frac{\exp(\sigma(x,x^-))}{\exp(\sigma(x,x^+))+\exp(\sigma(x,x^-))}\right]$$

Define $\widetilde{p} = p(\text{noisy label is correct})$. Note that

$$p(h(x)=h(\widetilde{x}^+), h(x)\neq h(\widetilde{x}^-)) = \widetilde{p}^3 + (1-\widetilde{p})^3$$

(i.e. all three points are correct or all reversed, such that their relative pairings are correct). In addition, the other three probabilities above are all equal to $\widetilde{p}(1-\widetilde{p})$. □

We now show that there exists a weighted loss function that in expectation equals $L_{triplet}$.

Lemma 6. *Define*

$$\widetilde{L}_{triplet} = \mathbb{E}_{\substack{x\sim\mathcal{P}, \\ \widetilde{x}_1^+, \widetilde{x}_2^+ \sim \widetilde{\mathcal{P}}^+(\cdot|x) \\ \widetilde{x}_1^-, \widetilde{x}_2^- \sim \widetilde{\mathcal{P}}^-(\cdot|x)}}\Bigg[-w^+\sigma(x,\widetilde{x}_1^+) + w^-\sigma(x,\widetilde{x}_1^-) + w_1\log\Big(\exp\Big(\sigma(x,\widetilde{x}_1^+)\Big) + \exp\Big(\sigma(x,\widetilde{x}_1^-)\Big)\Big)$$
$$- w_2\log\Big((\exp(\sigma(x,\widetilde{x}_1^+)) + \exp(\sigma(x,\widetilde{x}_2^+)))\cdot(\exp(\sigma(x,\widetilde{x}_1^-)) + \exp(\sigma(x,\widetilde{x}_2^-)))\Big)\Bigg]$$

where

$$w^+ = \frac{\widetilde{p}^2+(1-\widetilde{p})^2}{(2\widetilde{p}-1)^2} \qquad w^- = \frac{2\widetilde{p}(1-\widetilde{p})}{(2\widetilde{p}-1)^2} \qquad w_1 = \frac{\widetilde{p}^2+(1-\widetilde{p})^2}{(2\widetilde{p}-1)^2} \qquad w_2 = \frac{\widetilde{p}(1-\widetilde{p})}{(2\widetilde{p}-1)^2}$$

Then, $\mathbb{E}\left[\widetilde{L}_{triplet}\right] = L_{triplet}$.

Proof. We evaluate $\mathbb{E}\left[-w_1\sigma(x,\widetilde{x}_1^+)+w_2\sigma(x,\widetilde{x}_1^-)\right]$ and the other terms separately. Using the same probabilities as computed in Lemma 5,

$$\mathbb{E}\left[-w_1\sigma(x,\widetilde{x}_1^+)+w_2\sigma(x,\widetilde{x}_1^-)\right] = -(\widetilde{p}^2+(1-\widetilde{p})^2)w_1\mathbb{E}\left[\sigma(x,x_1^+)\right] - 2\widetilde{p}(1-\widetilde{p})w_1\mathbb{E}\left[\sigma(x,x_1^-)\right]$$
$$+ (\widetilde{p}^2+(1-\widetilde{p})^2)w_2\mathbb{E}\left[\sigma(x,x_1^-)\right] + 2\widetilde{p}(1-\widetilde{p})w_2\mathbb{E}\left[\sigma(x,x_1^+)\right]$$
$$= -\mathbb{E}\left[\sigma(x,x_1^+)\right]$$

We evaluate the remaining terms:

$$\mathbb{E}\left[w_3\log\left(\exp\left(\sigma(x,\widetilde{x}_1^+)\right)+\exp\left(\sigma(x,\widetilde{x}_1^-)\right)\right)\right] =$$
$$(\widetilde{p}^2+(1-\widetilde{p})^2)w_3\mathbb{E}\left[\log\left(\exp\left(\sigma(x,x_1^+)\right)+\exp\left(\sigma(x,x_1^-)\right)\right)\right]$$
$$+\widetilde{p}(1-\widetilde{p})w_3\mathbb{E}\left[\log\left((\exp(\sigma(x,\widetilde{x}_1^+))+\exp(\sigma(x,\widetilde{x}_2^+)))\cdot(\exp(\sigma(x,\widetilde{x}_1^-))+\exp(\sigma(x,\widetilde{x}_2^-)))\right)\right].$$
$$\mathbb{E}\left[w_4\log\left(\exp\left(\sigma(x,\widetilde{x}_1^+)\right)+\exp\left(\sigma(x,\widetilde{x}_2^+)\right)\right)\right] + \mathbb{E}\left[w_4\log\left(\exp\left(\sigma(x,\widetilde{x}_1^-)\right)+\exp\left(\sigma(x,\widetilde{x}_2^-)\right)\right)\right] =$$
$$(\widetilde{p}^2+(1-\widetilde{p})^2)w_4\mathbb{E}\left[\log\left(\exp\left(\sigma(x,x_1^+)\right)+\exp\left(\sigma(x,x_2^+)\right)\right)\right]$$
$$+4\widetilde{p}(1-\widetilde{p})w_4\mathbb{E}\left[\log\left(\exp\left(\sigma(x,x_1^+)\right)+\exp\left(\sigma(x,x_1^-)\right)\right)\right]$$
$$+((1-\widetilde{p})^2+\widetilde{p}^2)w_4\mathbb{E}\left[\log\left(\exp\left(\sigma(x,x_1^-)\right)+\exp\left(\sigma(x,x_2^-)\right)\right)\right]$$

Examining the coefficients, we see that

$$(\widetilde{p}^2+(1-\widetilde{p})^2)w_3 - 4\widetilde{p}(1-\widetilde{p})w_4 = \frac{(\widetilde{p}^2+(1-\widetilde{p})^2)^2}{(2\widetilde{p}-1)^2} - \frac{4\widetilde{p}^2(1-\widetilde{p})^2}{(2\widetilde{p}-1)^2} = 1$$
$$\widetilde{p}(1-\widetilde{p})w_3 - (\widetilde{p}^2+(1-\widetilde{p})^2)w_4 = \frac{\widetilde{p}(1-\widetilde{p})(\widetilde{p}^2+(1-\widetilde{p})^2)}{(2\widetilde{p}-1)^2} - \frac{(\widetilde{p}^2+(1-\widetilde{p})^2)\widetilde{p}(1-\widetilde{p})}{(2\widetilde{p}-1)^2} = 0$$

which shows that only the term $\mathbb{E}\left[\log\left(\exp\left(\sigma(x,x_1^+)\right)+\exp\left(\sigma(x,x_1^-)\right)\right)\right]$ persists. This completes our proof.

□

We now show the general case for debiasing $L_{attract}$.

Lemma 7. *Define $m = n+1$ (as the "batch size" in the denominator), and*

$$\widetilde{L}_{attract} = \mathbb{E}_{\substack{x\sim\mathcal{P}\\ \{\widetilde{x}_i^+\}_{i=1}^m\\ \{\widetilde{x}_j^-\}_{j=1}^m}}\Bigg[-w^+\sigma(x,\widetilde{x}_1^+)+w^-\sigma(x,\widetilde{x}_1^-) \tag{17}$$
$$+\sum_{k=0}^{m} w_k\log\left(\sum_{i=1}^{k}\exp\left(\sigma(x,\widetilde{x}_i^+)\right)+\sum_{j=1}^{m-k}\exp\left(\sigma(x,\widetilde{x}_j^-)\right)\right)\Bigg] \tag{18}$$

w^+ and w^- are defined in the same was as before. $\vec{w} = \{w_0,\dots w_m\}\in\mathbb{R}^{m+1}$ is the solution to the system $\mathbf{P}w = \mathbf{e}_2$ where $\mathbf{e}_2$ is the standard basis vector in $\mathbb{R}^{m+1}$ where the 2nd index is 1 and all others are 0. The i,jth element of $\mathbf{P}$ is $\mathbf{P}_{ij} = \widetilde{p}\mathbf{Q}_{i,j} + (1-\widetilde{p})\mathbf{Q}_{m-i,j}$ where

$$\mathbf{Q}_{i,j} = \begin{cases}\sum_{k=0}^{\min\{j,m-i\}}\binom{j}{k}\binom{m-j}{i-j+k}(1-\widetilde{p})^{i-j+2k}\widetilde{p}^{m+j-i-2k} & j\le i\\ \sum_{k=0}^{\min\{i,m-j\}}\binom{m-j}{k}\binom{j}{j-i+k}(1-\widetilde{p})^{j-i+2k}\widetilde{p}^{m-j+i-2k} & j>i\end{cases}$$

Then, $\mathbb{E}\left[\widetilde{L}_{attract}\right] = L_{attract}$.

We do not present the proof for Lemma 7, but the steps are very similar to the proof for the triplet loss case. We also note that a different from of $\mathbb{E}\left[\widetilde{L}_{attract}\right]$ must be computed for the multi-class case, which we do not present here (but can be derived through computation).

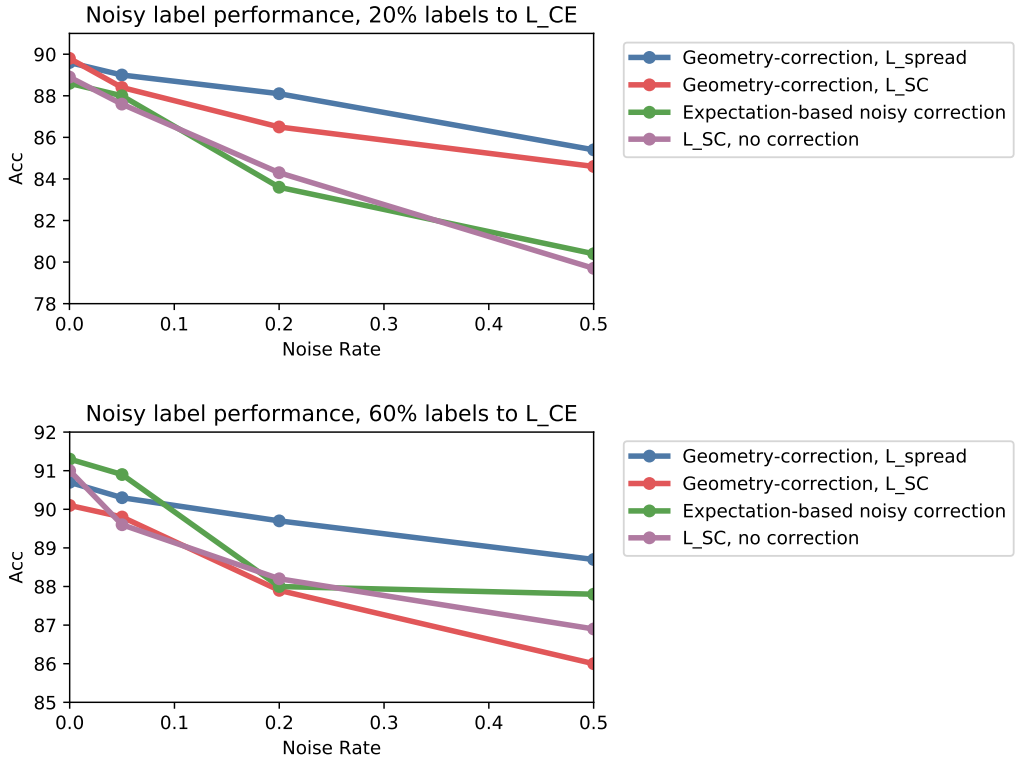

Figure 7: Performance of models under various amounts of label noise for the contrastive loss head, and various amounts of clean training data for the cross entropy loss.

Observation 4. *Note that the values of* $\mathbf{Q_{i,j}}$ *have high variance in the noise rate as* m *increases. Also note that the number of terms in the summation of* $\mathbf{Q_{i,j}}$ *increase combinatorially with* m*. We found this de-noising algorithm very unstable as a result.*

F.3.2 Additional Noisy Label Results

Now we report the performance of denoising algorithms with additional amounts of labeled data for the cross entropy loss head. We also report the performance of using $\widetilde{L}_{attract}$ to debias noisy labels.

Figure 7 shows the results. Our geometric correction together with L_{spread} works the most consistently. Using the geometric correction with L_{SC} can be unreliable, since L_{SC} can learn memorize noisy labels early on in training. The expectation-based debiasing algorithm $\widetilde{L}_{attract}$ occasionally shows promise but is unreliable, and is very sensitive to having the correct noise rate as an input.