WHAT CONTRASTIVE LEARNING LEARNS BEYOND CLASS-WISE FEATURES?

Anonymous authors
Paper under double-blind review

ABSTRACT

In recent years, contrastive learning has achieved the performance that is comparable to supervised learning in representation learning. However, the transferability of different contrastive learning methods to downstream tasks often varies greatly. In this paper, we study the downstream generalization ability of two contrastive learning methods: SimCLR and Spectral Contrastive Learning (Spectral CL). We find that beyond class-wise features, contrastive learning also learns two types of features, which we call shared features and subclass features, which play an important role in model transferability. SimCLR learns more shared and subclass features than Spectral CL, resulting in better transferability. We theoretically and experimentally reveal the mechanism by which SimCLR can learn more diverse features than Spectral CL. Therefore, we propose a method called High-pass Spectral CL to improve the transferability and generalization of Spectral CL, which achieves better performance than SimCLR and Spectral CL.

1 INTRODUCTION

In recent years, contrastive learning is rapidly developed, achieving comparable performance to supervised learning in pre-training (Chen et al., 2020; He et al., 2020). Various contrastive learning methods have been proposed and have achieved similar linear evaluation accuracy on ImageNet (Chen et al., 2020; He et al., 2020; Zbontar et al., 2021; Dwibedi et al., 2021). However, these methods often significantly differ in their ability to generalize to different downstream tasks.

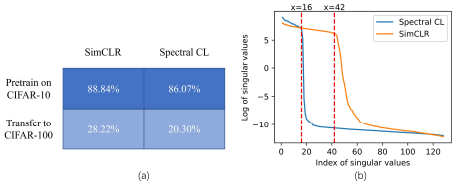

Figure 1: Comparison of SimCLR and Spectral CL on CIFAR-10. (a) Linear accuracies on pretraining and transferring datasets. (b) Singular value spectrum of the embedding space.

This paper explores the differences in downstream generalization between two classical contrastive learning methods, SimCLR (Chen et al., 2020) and Spectral Contrastive Learning (Spectral CL) (HaoChen et al., 2021). We justify the reasons for choosing these two methods as follows: 1) the loss functions of SimCLR and Spectral CL have similar forms and both have been studied theoretically, yet, their differences are unknown experimentally; and 2) their training approach follows the simplest way without techniques like momentum encoder (He et al., 2020) or predictor (Grill et al., 2020), facilitating our discussion about the impact of loss on contrastive learning.

Through experiments, we find that although these two methods have similar linear evaluation accuracy on the pretraining dataset, their transferability differs significantly, as shown in Figure 1(a). The finding motivates us to explore the two methods more deeply to understand the impact of loss function design on downstream generalization. By analyzing the eigen-spectrums of the embedding vectors learned by these two methods, we find that the features learned by SimCLR are distributed in higher dimensional subspaces compared to Spectral CL, as shown in Figure 1(b). Intuitively, this means that SimCLR learns more diverse features than Spectral CL, thus achieving better transferability. In this paper, our goal is to understand the embedding space of contrastive learning and the impact of loss function design on the embedding space of contrastive learning.

2 INVESTIGATING FEATURES LEARNED WITH CONTRASTIVE LOSSES

In this section, we categorize the features learned by contrastive learning into three types: 1) **class-wise features** that correspond to class center vectors and reflect the clustering structure of the samples, 2) **shared features** that correspond to features common among different categories of samples, such as color and pose and 3) **subclass features** that correspond to the information of subclasses within each class. Our experiments show that class-wise features lead to good performance on pre-training datasets for contrastive learning, but do not guarantee good downstream generalization. And the latter two types of features are crucial for downstream generalization. We use toy models pretrained on CIFAR-10 to illustrate our points in this section.

2.1 CLASS-WISE FEATURES REPRESENTED BY CLASS CENTERS

In previous work (Arora et al., 2019; Wang et al., 2022), for ease of theoretical analysis, they usually adopted the mean classifier, which is defined below.

Definition 2.1 (Mean Classifier)**.** Given the learned features of training samples from class i, $z_1^i, \cdots, z_{n_i}^i$, the mean classifier uses the class center $\mu_i = \frac{1}{n_i}\sum_{k=1}^{n_i} z_k^i$ as the weight of the linear classifier W, i.e. $W = [\mu_1, ..., \mu_c]$.

Table 1: Test classification accuracies (%) for learned classifier and mean classifier on CIFAR-10. Models are pretrained on CIFAR-10.

Method	Learned classifier	Mean classifier
Spectral CL	86.15	82.97
SimCLR	88.56	86.05

Here, we conduct the experiments with the mean classifier and the learned classifier on the embedding vectors learned by SimCLR and Spectral CL on CIFAR-10. The results are shown in Table 1. The two classifiers achieve similar results on pretraining dataset, which suggests that the features obtained by projecting the embedding onto a subspace U_c of rank 10 spanned by the class centers are able to perform the linear classification well on the pretraining dataset. However, from Figure 1(b), we can see that both SimCLR and Spectral CL have embedding space with a rank higher than 10, suggesting that both methods extract many other features except classwise features. In the next section, we will focus on analyzing what additional features beyond the 10 class centers of CIFAR-10 that contrastive learning has learned.

2.2 BEYOND SEPARATION: SHARED AND SUBCLASS FEATURES

In this section, we investigate features in the orthogonal complementary space of U_c denoted as $U_c^\perp$. We classify these features into two categories: shared features and subclass features. To extract the shared features and the subclass features, we resort to the concept of principal angles and vectors, which are the generalization of the concept of "angle" in three-dimensional space. In Appendix D, we introduce the concept and calculation of principal angles and vectors in detail.

Here we study the distribution of embedding vectors learned by contrastive learning on $U_c^\perp$. And we denote the projection operator onto a subspace V as $P_V(\cdot)$. Given the embedding vector z, we project z onto $U_c^\perp$ and denote the projection as $p = P_{U_c^\perp}(z)$. To study the shared features and subclass features of class i, we calculate the principal angles and vectors between the subspace spanned by the projections p from class i and the subspace spanned by the projections p from other classes. See the implementation details in Appendix D.

Shared Features. In Figure 2(a), we can observe that most of the principal angles between the subspaces are concentrated around 0 degrees. The corresponding principal vectors are referred as shared features since the embedding spaces from different classes are parallel in these directions. In Figure 2(c), we can see that these features often correspond to features shared between classes such as color and pose information.

Subclass Features. In addition to shared features, there is also a portion of principal angles that are close to 90 degrees (Figure 2(a)). The corresponding principal vectors are denoted as subclass features since only one class significantly distributes in this direction, while the embedding spaces of other classes are vertical to this direction. In Figure 2(b), we can see that these features correspond to subclass information.

Comparison between SimCLR and Spectral CL. Figure 2(a) indicates that SimCLR learns more shared and subclass features compared to Spectral CL. This suggests that SimCLR learns more

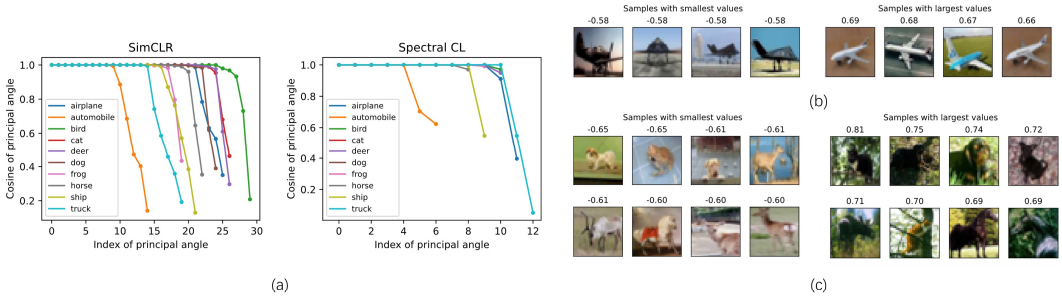

Figure 2: Experiment on the principal angles and vectors. (a) shows the principal angles of SimCLR and Spectral CL. (b) and (c) respectively visualize an instance of subclass and shared features of SimCLR. The values over images depict the projection length on the feature direction. For the subclass (shared) feature, we visualize 4 (8) samples with minimum and maximum values respectively. In (b), the subclass feature is unique to the airplane class, distinguishing between fighters and airliners. In (c), the shared feature reflects the shade of the animal's color.

diverse features than Spectral CL. We believe that this is the key reason for SimCLR's superior downstream generalization performance compared to Spectral CL, as verified in Appendix B.2.

3 Pushing Contrastive Learning to Learn Diverse Features

In this section, we theoretically explore the mechanism by which SimCLR learns higher dimensional embedding than Spectral CL and propose a general method, High-pass Spectral Contrastive Learning (HSCL), which indeed improves the generalization over the vanilla Spectral CL.

3.1 Notation and Formulation

Data Generation. Following HaoChen et al. (2021), given a set of natural data $\bar{\mathcal{X}} = \{\bar{x} | \bar{x} \in \mathbb{R}^d\}$, we draw a natural example $\bar{x}$ as distribution $\mathcal{P}_d(\bar{x})$. And then draw the positive pairs x, x^+ from random augmentation on $\bar{x}$ with distribution $\mathcal{A}(\cdot|\bar{x})$. We denote the collection of augmented views as $\mathcal{X}$. Considering the N samples in $\mathcal{X}$ as N nodes, we construct an augmentation graph $\mathcal{G} = (\mathcal{X}, A)$ with an adjacency matrix A. The edeg weight $A_{xx'}$ between x and x' is defined as the joint probability $A_{xx'} = \mathbb{E}_{\bar{x}} \mathcal{A}(x|\bar{x})\mathcal{A}(x'|\bar{x})$. And we denotes the diagnol degree matrix as $D = deg(A)$,i.e., $D_{xx} = w_x = \sum_{x'} A_{xx'}$.

Formulation. Here we formulate the contrastive loss into the expected form. Given the positive pair (x, x^+) generated by data augmentations, and independently sampled negative samples x', we learn an encoder $f : \mathbb{R}^d \to \mathbb{R}^m$. We formulate the InfoNCE loss and spectral contrastive loss as follows:

$$\begin{aligned} \mathcal{L}_{nce} &= -\mathbb{E}_{x,x^+}[f(x)^\top f(x^+)] + \mathbb{E}_x \log \mathbb{E}_{x'}[\exp(f(x)^\top f(x'))], \\ \mathcal{L}_{sp} &= -\mathbb{E}_{x,x^+}[f(x)^\top f(x^+)] + \frac{1}{2}\mathbb{E}_x \mathbb{E}_{x'}[(f(x)^\top f(x'))^2]. \end{aligned} \tag{1}$$

And we decompose the contrastive loss $\mathcal{L}_{cl}$ into two parts, the alignment loss $\mathcal{L}_{align}$ and the uniformity loss $\mathcal{L}_{unif}$, i.e., $\mathcal{L}_{cl} = \mathcal{L}_{align} + \mathcal{L}_{unif}$, where, $\mathcal{L}_{align} = -\mathbb{E}_{x,x^+}[f(x)^\top f(x^+)]$, $\mathcal{L}_{unif}^{(nce)} = \mathbb{E}_x \log \mathbb{E}_{x'}[\exp(f(x)^\top f(x'))]$, $\mathcal{L}_{unif}^{(sp)} = \frac{1}{2}\mathbb{E}_x \mathbb{E}_{x'}[(f(x)^\top f(x'))^2]$.

3.2 Dynamics of Singular Value Spectrum

In this section, we study the dynamics of the singular value spectrum of SimCLR and Spectral CL. We first study the dynamics of embedding matrix F via gradient flow, where $F \in \mathbb{R}^{m \times N}$ is an embedding matrix where each column represents an embedding vector of samples.

Theorem 3.1 (Dynamics of Embedding Matrix)**.** *The embedding matrix F evolves by:*

$$\dot{F} = -(G_{align} + G_{unif})$$

where G_{align} is the gradient of alignment loss and G_{unif} is the gradient of uniformity loss. And we have $G_{align} = -2FA$, $G_{unif} = 2FA'$, where, $A' = A'_{sp} = DF^T FD$ for Spectral CL and

$A' = A'_{nce} = (DD_{exp}^{-1}A_{exp} + A_{exp}D_{exp}^{-1}D)/2$ *for SimCLR, where* $A_{exp} = D\exp(F^TF)D$ *and* $D_{exp} = deg(A_{exp})$. *Note that,* $\log$, $\exp$ *are element-wise operations here.*

More specifically, the dynamic of the singular value of F obeys the following property.

Theorem 3.2 (Dynamics of Singular Value)**.** *With a fixed* $\Sigma = A - A'$, *if* Σ *has negative eigenvalues, the embedding matrix* F *has vanishing singular values.*

Here, we record the evolution of singular values of embedding space of SimCLR and Spectral CL during the training in Figure 3. It can be seen that a large portion of singular values of SimCLR rise during the training process, while Spectral CL has only a few singular values rising. This implies that $A - A'_{nce}$ has more positive eigenvalues than $A - A'_{sp}$, and explains the better feature diversity of SimLCR than Spectral CL after training (Figure 1(b)).

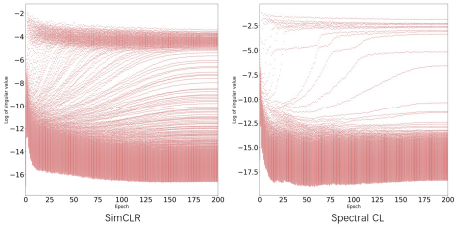

Figure 3: The evolution of singular values of embedding space during the training process of SimCLR and Spectral CL.

For ease of discussion, we make an assumption on the alignment between matrix A and A'.

Assumption 3.3 (Eigenspace Alignment)**.** We assume that A, A'_{nce} and A'_{sp} have the same eigenspaces during the training, i.e., $\exists V$, s.t. $A = V\Lambda_d V^\top$, $A'_{nce} = V\Lambda_{nce}V^\top$ and $A'_{sp} = V\Lambda_{sp}V^\top$, where Λ_d, Λ_{nce} and Λ_{sp} are diagonal matrices consisting of eigenvalues λ_d^i, λ_{nce}^i and λ_{sp}^i.

The assumption is a natural consequence under the interpretation of contrastive learning as spectral decomposition of the adjacency matrix A by HaoChen et al. (2021). Under Assumption 3.3, the eigenvalues of $A - A'$ can be easily computed by $\lambda_d^i - \lambda_a^i$, where λ_a^i is the eigenvalue of A'. By Theorem 3.1, to keep more singular values of embedding space growing, we need to make $\lambda_d^i - \lambda_a^i$ greater than zero. Actually, we verify that SimCLR achieves this by applying a high-pass filter function on A'_{sp} to lower the eigenvalues λ_a^i in Appendix B.3.

3.3 Proposed Method: High-pass Spectral CL

Inspired by the mechanism of SimCLR applying a high-pass filter on A'_{sp}, we propose a method called High-pass Spectral CL (HSCL) to improve the performance of Spectral CL. We directly apply a high-pass filter function on the uniformity term of spectral contrastive loss.

$$\begin{aligned}\mathcal{L}_{HSCL} = &- \mathbb{E}_{x,x^+}[f(x)^\top f(x^+) \\ &+ \frac{1}{2}\mathbb{E}_x\mathbb{E}_{x'}[(f(x)^\top f(x'))(Wf(x))^\top(Wf(x'))],\end{aligned} \tag{2}$$

where W is a high-pass filter on eigenspace of embedding, i.e., $W = Ug(\Lambda)U^\top$ where U and Λ are from the SVD decomposition on matrix $FD = U\Lambda V^\top$ and $g(\cdot)$ is a high-pass filter function. In Appendix C.1, we show the pseudocode of our algorithm. We verify the effectiveness of our method by experiment in Appendix C.3.

Here we show that applying a high-pass filter on the uniformity loss is equal to applying a high-pass filter on A'_{sp}.

Theorem 3.4 (HSCL Applys a High-pass Filter on A'_{sp})**.** *HSCL implicitly applys the high-pass function on* A'_{sp}, *i.e.,* $\lambda_{HSCL}^i = g(\lambda_{sp}^i)\lambda_{sp}^i$, *where* $g(\cdot)$ *is a high-pass filter function, and* λ_{HSCL}^i *is the eigenvalue of* A'_{HSCL}.

4 Conclusion

In this paper, we discuss the features learned by contrastive learning and the impact of these features on downstream generalization. We show that in addition to class-wise features, contrastive learning also learns shared features and subclass features. SimCLR achieves better downstream generalization and transferability than Spectral CL by learning more shared features and subclass features. Meanwhile, we theoretically reveal the mechanism that SimCLR achieves learning to more diverse features is to apply a high-pass function on the matrix A'. We hope that our work could inspire more studies about the design and interpretation of contrastive loss functions.

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.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9588–9597, 2021.

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

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9598–9608, 2021.

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

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

Yifei Wang, Qi Zhang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. *arXiv preprint arXiv:2203.13457*, 2022.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

A Background and Related Work

Contrastive learning and understandings. Contrastive learning has achieved great success and shown comparable performance to supervised learning in learning visual representation. A widely adopted learning objective is the InfoNCE loss (Oord et al., 2018), with representative methods like SimCLR (Chen et al., 2020) and MoCo (He et al., 2020). HaoChen et al. (2021) recently proposed a new contrastive loss, named spectral contrastive loss, that also achieves comparable performance on downstream data. Meanwhile, they draw a formal connection between this objective and a matrix decomposition problem, based on which they establish guarantees on downstream data.

Dimensional collapse of contrastive learning. Dimensional collapse (Hua et al., 2021) is a common phenomenon in contrastive methods, where the embedding only spans a low dimensional subspace. Recent work (Jing et al., 2021) analyses the dynamics of contrastive learning and points out that strong data augmentation and implicit regularization drives models toward low-rank solutions.

B Additional Experiment

B.1 Visualization of Shared Features and Subclass Features

Here we visualize shared features and subclass features of Spectral CL and SimCLR in Figure 4, 5, 6, 7.

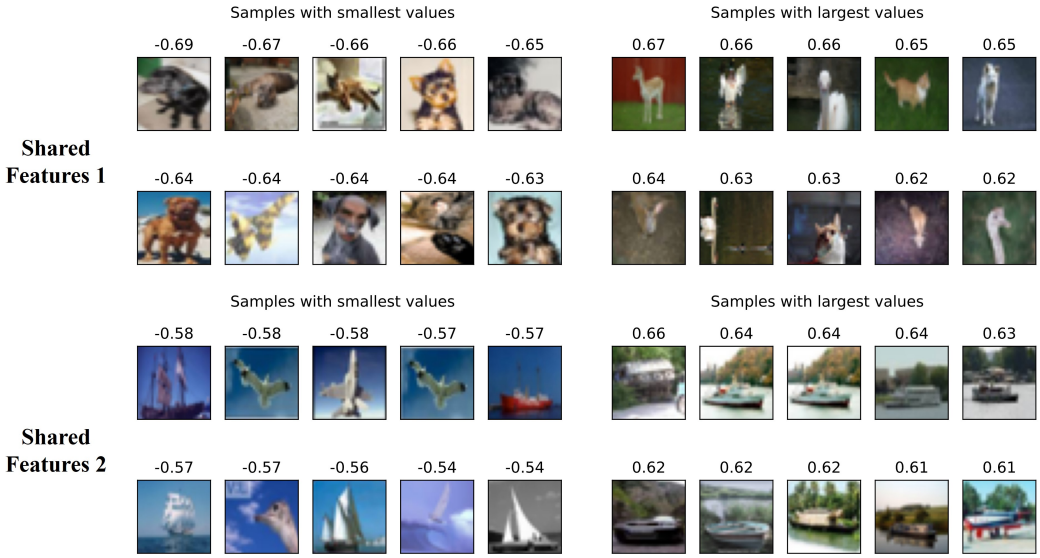

Figure 4: Two instances of shared features of SimCLR trained on CIFAR-10.

B.2 Shared and Subclass Features are Crucial to Downstream Tasks

Models learned by self-supervised learning are often utilized as feature extractors to transfer to other downstream tasks. In order to measure the generalization ability of the model, we transfer a pre-trained model on CIFAR-10 to CIFAR-100. To demonstrate the impact of shared features and subclass features on the generalization ability of the model, we project the extracted embedding vectors onto subspaces spanned by different kinds of features and train a linear classifier on the projected vectors to obtain classification accuracy.

Experimental results in Table 2 indicate that utilizing class centers demonstrate good classification performance on the pre-training dataset CIFAR-10 and the enhancement in classification accuracy derived from utilizing shared and subclass features is relatively modest. Conversely, when evaluated on the transfer dataset CIFAR-100, the enhancement in classification accuracy derived from utilizing

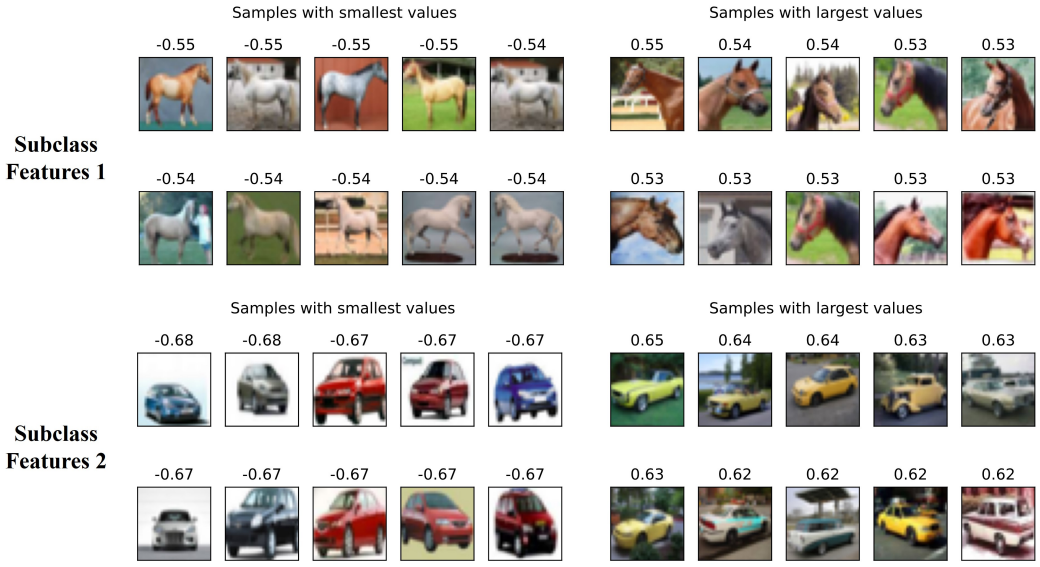

Figure 5: Two instances of subclass features of SimCLR trained on CIFAR-10.

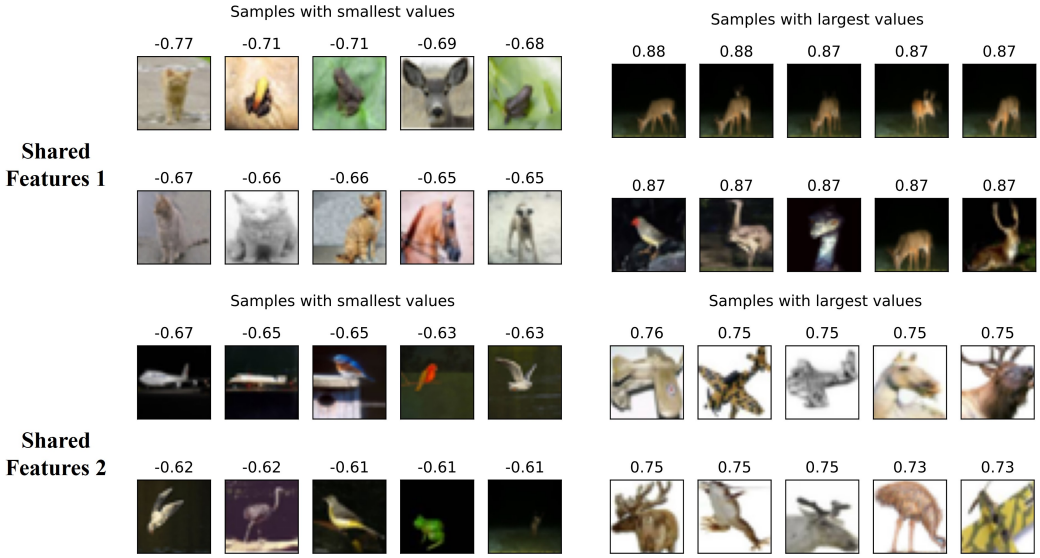

Figure 6: Two instances of shared features of Spectral CL trained on CIFAR-10.

shared and subclass features is substantial (with the exception of Spectral CL's subclass features, which did not yield a considerable enhancement due to a dearth of subclass features). This suggests that achieving good performance on a pre-training dataset does not ensure good transferability to downstream tasks. Incorporating a more diverse set of features, comprising both shared and subclass features can augment the model's generalization capability.

B.3 SimCLR Applys a High-pass Filter on A'_{sp}

As discussed in Section 3.2, SimCLR has more singular values rising than Spectral CL, which suggests that $A - A'_{nce}$ has more positive eigenvalues than $A - A'_{sp}$. Here we experimently show that SimCLR applys a high-pass filter on A'_{sp}, i.e., $g(\lambda^i_{sp}) = \lambda^i_{nce}/\lambda^i_{sp}$ is a high-pass spectral filter. For clarity, we state the definition of spectral filter below.

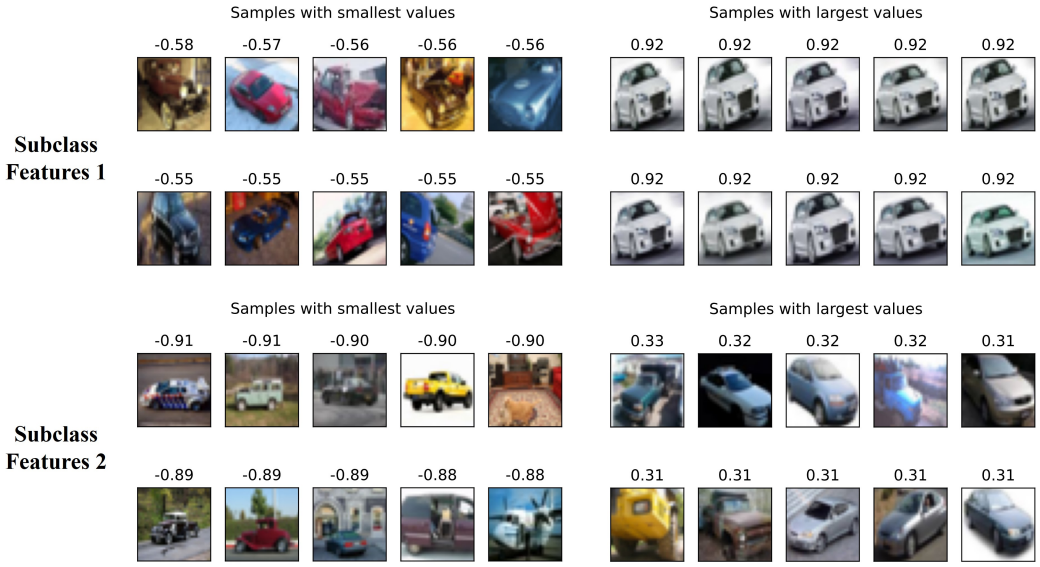

Figure 7: Two instances of subclass features of Spectral CL trained on CIFAR-10.

Table 2: Linear classification accuracy on CIFAR-10 and CIFAR-100 after projecting embeddings into different subspaces. U_c, U_b, U_s respectively represent subspaces spanned by class centers, subclass features, and shared features. The model is pretrained on CIFAR-10.

Method	Subspace	CIFAR-10	CIFAR-100
SimCLR	U_c (baseline)	86.73	17.58
	$U_c + U_b$	87.41 (+0.68)	21.61 (+4.03)
	$U_c + U_s$	89.19 (+2.46)	31.74 (+14.16)
Spectral CL	U_c (baseline)	85.78	18.55
	$U_c + U_b$	85.96 (+0.18)	18.91 (+0.36)
	$U_c + U_s$	86.37 (+0.59)	23.64 (+5.09)

Definition B.1 (Spectral Filter)**.** A spectral filter process G of a signal f is to apply a scalar function (i.e. a spectral filter) $g : \mathbb{R} \to \mathbb{R}$ element-wisely on its eigenvalues in its spectral domain, i.e., $u_x = Gf(x) = Vg(\Lambda)V^\top f(x)$, where $G = Vg(\Lambda)V^\top$ is also called a spectral convolution operator. A filter can be categorized as low-pass or high-pass. Generally speaking, a high-pass filter will lower the large eigenvalues and amplify small eigenvalues, i.e., a high-pass filter is a monotonically decreasing function, while a low-pass filter does the opposite.

SimCLR Applys a High-pass Filter on A'_{sp}. To show how SimCLR keeps the singular values growing, we investigate the spectral filter $g(\lambda^i_{sp}) = \lambda^i_{nce}/\lambda^i_{sp}$. From Figure 8, we can see that the filter function lowers the higher eigenvalues such that SimCLR can keep $\lambda^i_d - \lambda^i_a > 0$ for a large portion of eigenvalues.

C Experiment

C.1 Algorithm

Here we show the pseudocode of HSCL in Algorithm 1.

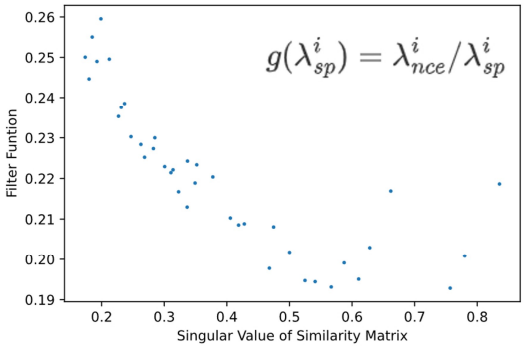

Figure 8: Spectral filter function $g(\lambda^i_{sp}) = \lambda^i_{nce}/\lambda^i_{sp}$. A'_{nce} and A'_{sp} are computed on the embeddings learned by SimCLR on CIFAR-10.

Algorithm 1 Pseudocode of HSCL

Require: batch size N, structure of encoder network f, filter function g
for sampled minibatch $\{\bar{x}_i\}_{i=1}^N$ **do**
  for $i \in \{1, \cdots, N\}$ **do**
    draw two augmentations $x_i = \mathrm{aug}(\bar{x}_i)$ and $x'_i = \mathrm{aug}(\bar{x}'_i)$
    compute $z_i = f(x_i)$ and $z'_i = f(x'_i)$.
  end for
  compute matrix $B = \sum_{i=1}^N (z_i z_i^\top + z'_i z'^\top_i)$.
  Apply eigen-decomposition $B = VSV^\top$.
  Compute filter matrix $W = Vg(S^{1/2})V^\top$.
  Compute loss $\mathcal{L} = -\frac{2}{N}\sum_{i=1}^N z_i^\top z'_i + \frac{1}{N(N-1)}\sum_{i\neq j}(z_i^\top z'_j)((Wz_i)^\top(Wz'_j))$.
  update f to minimize $\mathcal{L}$.
end for

C.2 Implementation Details

Pretraining. We pretrain the model on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009). For the backbone network, we use the CIFAR variant of ResNet18 (He et al., 2016). For the projection head, we use a 2-layer MLP with output dimensions 128 with ReLU. We train the model with SGD with momentum 0.9 and weight decay 1×10^{-5}. The learning rate starts at 0.3 with linear warmup for the first 10 epochs and decreases to 0 with cosine decay schedule. We train for 1000 epochs with batch size 256. For HSCL, we choose variants with three different high-pass filter functions for our experiments: $g(\lambda) = \lambda^{-0.1}$, $g(\lambda) = \lambda^{-0.3}$ and $g(\lambda) = \lambda^{-0.5}$.

Linear Evaluation. We train a linear classifier on the embedding vectors using SGD with momentum 0.9 and batch size 64 for 100 epochs. We run the experiments with learning rate starting from 1, 0.1, 0.01 and decayed by $10\times$ at the 60th and 80th epochs. We report the best linear accuracy.

Transfer Learning. To evaluate the transferability of the models, we transfer models pretrained on CIFAR-10 to CIFAR-100 by training a linear classifier on the embedding vectors. The settings for training the linear classifier are the same as in linear evaluation.

C.3 Results

HSCL Learns Higher Dimensional Features. We compute the singular value spectrum of the embedding space learned by HSCL on CIFAR-10. From Figure 9, it can be seen that HSCL learns a higher dimensional representation than Spectral CL.

HSCL Achieves Better Generalization. We report the accuracy on CIFAR-10/100 in Table 3. The experiment results show that the embeddings learned by HSCL achieve better performance than SimCLR and Spectral CL. In Table 4, we report the accuracy of transferring the model pretraining on CIFAR-10 to CIFAR-100, and show that our algorithm has better transferability than SimCLR and Spectral CL.

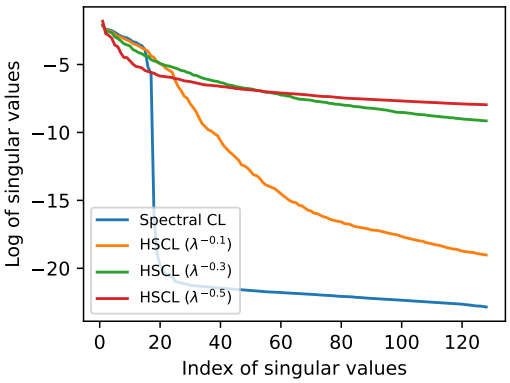

Figure 9: Singular value spectrums of the embedding spaces learned on CIFAR-10. Obviously, HSCL has higher dimensional embedding spaces than Spectral CL.

Table 3: Accuracies of linear classifiers trained on embeddings learned with different methods on pretraining datasets CIFAR-10 and CIFAR-100.

Method	CIFAR-10	CIFAR-100
Spectral CL	86.07	47.76
SimCLR	88.84	56.77
HSCL ($g(\lambda) = \lambda^{-0.1}$)	88.46	53.10
HSCL ($g(\lambda) = \lambda^{-0.3}$)	**90.56**	59.83
HSCL ($g(\lambda) = \lambda^{-0.5}$)	90.42	**61.91**

D Implementation Details about Shared Features and Subclass Features.

D.1 Principle Angles and Vectors

Here we introduce the concept of principle angles and vectors.

Definition D.1 (Principal angles and vectors)**.** Consider two subspaces X,Y in the n-dimentional eucledian space R^n of dimensions p and d, respectively. The $m = min(p, q)$ principal angles $0 \leq \theta_1 \leq \theta_2 \leq ... \leq \theta_m \leq \pi/2$ between these subspaces, and their corresponding principal vectors, are defined recursively by

$$\cos(\theta_k) = \max_{x \in X, y \in Y} |x^\intercal y| = |x_k^\intercal y_k|$$

subject to

$$\|x\| = \|y\| = 1, x^\intercal x_i = 0, y^\intercal y_i = 0, i = 1, ..., k-1$$

The vectors $\{x_1, ..., x_m\}$ and $\{y_1, ..., y_m\}$ are the principal vectors.

And here we introduce the computation of principal angles and vectors.

Theorem D.2 (Computation of Principal Angles and Vectors)**.** *Let the columns of matrices $X \in \mathbb{R}^{n \times p}$ and $Y \in \mathbb{R}^{n \times q}$ form orthonormal bases for the subspaces $\mathcal{X}$ and $\mathcal{Y}$, correspondingly. Let the SVD of $X^\top Y$ be $U \Sigma V^\top$, where U and V are unitary matrices and $Sigma$ is a $p \times q$ diagonal matrix with diagonal elements $s_1(X^\top Y)$,...,$s_m(X^\top Y)$ in nonincreasing order with $m = \min(p, q)$. Then*

$$\cos(\theta_k) = s_k(X^\top Y)$$

where θ_k denotes the k-th principal angle between $\mathcal{X}$ and $\mathcal{Y}$. And the princpal vectors associated with this pair of subspaces are given by the first m columns of XU and YV, correspondingly.

D.2 Separation of Shared Features and Subclass Features.

Here we show the implementation details about the separation of shared features and subclass features. To get the shared features and subclass features of class i, we first collect the projection belonging to

Table 4: Accuracies of transferring models pretrained on CIFAR-10 to CIFAR-100.

Method	CIFAR-100
Spectral CL	20.30
SimCLR	28.22
HSCL ($g(\lambda) = \lambda^{-0.1}$)	24.56
HSCL ($g(\lambda) = \lambda^{-0.3}$)	33.85
HSCL ($g(\lambda) = \lambda^{-0.5}$)	**36.84**

class i, $P_i = [p_1^i, ..., p_{n_i}^i]$, and the projection belonging to other classes, $\bar{P}_i = \cup^{j \neq i} P_j$. We denote the subspace spanned by vectors in P_i as V_i and the subspace spanned by vectors in $\bar{P}_i$ as $\bar{V}_i$.

Following Theorem D.2, we first calculate the orthonormal bases for the subspaces V_i and $\bar{V}_i$. We show the calculation process here as an example for bases of V_i. Apply SVD decomposition on P_i, and get $P_i = U\Lambda V^T$, where the diagonal elements of Λ are λ_1,...,λ_d in descending order. We select the first m column of U as the bases for V_i, where $m = \min\{k | \frac{\sum_{j=1}^{k} \lambda_j^2}{\sum_{j=1}^{d} \lambda_j^2} > 0.995\}$. The process of calculating the bases for $\bar{V}_i$ is the same.

Then follow Theorem D.2, we can compute the principal angles and vectors between V_i and $\bar{V}_i$.

E Omitted Proofs

In this section, we present proofs for all theorems in the main paper.

E.1 Proof of Theorem 3.1

First, we reformulate the alignment loss and the uniformity loss into matrix form:

$$\begin{aligned} \mathcal{L}_{align} &= -Tr(FAF^T) \\ \mathcal{L}_{unif}^{(sp)} &= \frac{1}{2}\|D^{1/2}F^TFD^{1/2}\|_F^2 \\ \mathcal{L}_{unif}^{(nce)} &= Tr(D\log(\deg(\exp(F^TF)D))). \end{aligned} \tag{3}$$

Then we derive the gradient of the alignment loss and the uniformity loss.

The gradient of $\mathcal{L}_{align}$:

$$\begin{aligned} \frac{\partial \mathcal{L}_{align}}{\partial F} &= -\frac{\partial \mathrm{Tr}(FAF^\top)}{\partial F} \\ &= -(FA^\top + FA) \\ &= -2FA \end{aligned}$$

The gradient of $\mathcal{L}_{unif}^{(sp)}$:

$$\begin{aligned} \frac{\partial \mathcal{L}_{unif}^{(sp)}}{\partial F} &= -\frac{\partial \frac{1}{2}\|D^{1/2}F^TFD^{1/2}\|_F^2}{\partial F} \\ &= 2F(DF^\top FD) \end{aligned}$$

The gradient of $\mathcal{L}_{unif}^{(nce)}$ on $f(x)$:

$$\begin{aligned} \frac{\partial \mathcal{L}_{unif}^{(HSCL)}}{\partial f(x)} &= \frac{\partial \sum_x w_x \log \sum_{x'} w_{x'}[\exp(f(x)^\top f(x'))]}{\partial f(x)} \\ &= \sum_{x'} (w_{x'} \frac{w_x \exp(f(x')^T f(x))}{\sum_i w_{x_i} \exp(f(x')^T f(x_i))} + w_x \frac{w_{x'} \exp(f(x)^T f(x'))}{\sum_j w_{x_j} \exp(f(x)^T f(x_j))}) f(x') \end{aligned}$$

In matrix, we have $\frac{\partial \mathcal{L}_{unif}^{(HSCL)}}{\partial F} = 2FA'_{nce}$.

E.2 PROOF OF THEOREM 3.2

Proof. Here we follow the proof in Jing et al. (2021). According to Theorem 3.1, we have

$$\frac{d}{dt}F = 2F\Sigma$$

For a fixed Σ, we solve this equation analyically,

$$F(t) = F(0)\exp(2\Sigma t)$$

Apply eigen-decomposition on Σ, $\Sigma = U\Lambda U^\top$. Therefore,

$$F(t) = F(0)U\exp(2\Lambda t)U^\top$$

When Σ has negative eigenvalues, i.e., Λ has negative terms, we have for $t \to \infty$, $\exp(2\Lambda t)$ is rank deficient. Therefore, we know that $F(\infty)$ is also rank deficient, the embedding matrix F has vanishing singular values. □

E.3 PROOF OF THEOREM 3.4

The gradient of $\mathcal{L}_{unif}^{(HSCL)}$ on $f(x)$:

$$\begin{aligned}
\frac{\partial \mathcal{L}_{unif}^{(HSCL)}}{\partial f(x)} &= \frac{1}{2}\frac{\partial \mathbb{E}_x \mathbb{E}_{x'}[(f(x)^\top f(x'))(Wf(x))^\top(Wf(x'))]}{\partial f(x)} \\
&= \frac{1}{2}\frac{\partial \sum_x w_x \sum_{x'} w_{x'}(f(x)^\top f(x'))(Wf(x))^\top(Wf(x'))}{\partial f(x)} \\
&= 2\sum_{x'}(f(x)^\top W^\top W f(x'))f(x')
\end{aligned}$$

In matrix,

$$\frac{\partial \mathcal{L}_{unif}^{(HSCL)}}{\partial F} = 2F(DF^\top(W^\top W)FD)$$

So $A'_{HSCL} = DF^\top(W^\top W)FD$. Note that $FD = U\Lambda V^T$ and $W = Ug(\Lambda)U^\top$, we have,

$$\begin{aligned}
A'_{HSCL} &= V\Lambda U^\top(Ug(\Lambda)^2U^\top)U\Lambda V^\top \\
&= V\Lambda^2 g(\Lambda)^2 V^\top
\end{aligned}$$

Note that, $A'_{sp} = V\Lambda U^\top U\Lambda V^\top = V\Lambda^2 V^\top$. So we have

$$\lambda^i_{HSCL} = \lambda^i_{sp} g'(\lambda^i_{sp})$$

where $g'(\cdot)$ is a high-pass filter function defined as $g'(\lambda^i_{sp}) = (g(\Lambda)_i)^2$.