Hyperspherical Consistency Regularization

Cheng Tan\textsuperscript{1,2,*}, Zhangyang Gao\textsuperscript{1,2,*}, Lirong Wu\textsuperscript{1,2}, Siyuan Li\textsuperscript{1,2}, Stan Z. Li\textsuperscript{1,2}
\textsuperscript{1} AI Lab, School of Engineering, Westlake University
\textsuperscript{2} Institute of Advanced Technology, Westlake Institute for Advanced Study
\textsuperscript{*}\{tancheng,gaozhangyang,wulirong,lisiyuan,stan.zq.li\}@westlake.edu.cn

Abstract

Recent advances in contrastive learning have enlightened diverse applications across various semi-supervised fields. Jointly training supervised learning and unsupervised learning with a shared feature encoder becomes a common scheme. Though it benefits from taking advantage of both feature-dependent information from self-supervised learning and label-dependent information from supervised learning, this scheme remains suffering from bias of the classifier. In this work, we systematically explore the relationship between self-supervised learning and supervised learning, and study how self-supervised learning helps robust data-efficient deep learning. We propose hyperspherical consistency regularization (HCR), a simple yet effective plug-and-play method, to regularize the classifier using feature-dependent information and thus avoid bias from labels. Specifically, HCR first project logits from the classifier and feature projections from the projection head on the respective hypersphere, then it enforces data points on hyperspheres to have similar structures by minimizing binary cross entropy of pairwise distances’ similarity metrics. Extensive experiments on semi-supervised and weakly-supervised learning demonstrate the effectiveness of our method, by showing superior performance with HCR.

1. Introduction

The last decade has witnessed revolutionary advances in deep learning across various computer vision fields such as image classification \[26,29,38,72\], object detection \[48,64–66\], and semantic segmentation \[25,47,67\] in the presence of large-scale labeled datasets. However, massive collection and accurate annotation of datasets are time-consuming and expensive. In many practical situations, only small-scale high-quality labeled datasets are available. For this reason, semi-supervised learning (SSL) that learning from few labeled data and a large number of unlabeled data has received broad attention \[4,5,42,62,63,69,71,73,84,85\].

With the development of contrastive learning \[7,9–11,20,24,27,28,45,74,80,90,94\], recent SSL algorithms \[23,34,43,79,92\] tend to extend self-supervised learning into supervised learning by adding a branch network as a projection head that jointly learns from feature-dependent and label-dependent information. Though the feature encoder is supposed to learn better by making agreements from different views on latent spaces, the classifier which determines the ultimate predictions still suffers from the bias of semi-supervision or weak-supervision. Typically, \[33,93\] found that data imbalance is not the key issue in learning high-quality representations from long-tail data, while simply adjusting the classifier with balanced sampling can effectively alleviate the imblanced bias. This phenomenon suggests that decent representation may help but not be enough for robust learning, while regularizing the classifier is necessary to improve learning performance.
A vast number of current empirical contrastive learning methods [9–11, 20, 24, 28, 45] project feature embeddings on a hypersphere through $\ell_2$ normalization while maximizing distances between negative pairs and minimizing distances between positive pairs, as shown in Figure 1. Restricting the output space to a unit hypersphere can improve training stability in machine learning where dot products are ubiquitous [77, 80, 86]. Besides, well-clustered features on the hypersphere are linearly separable from the rest of the feature space. The above desirable traits are considered to be useful while regularizing the classifier.

In this work, we analyze the relationship between the projection head and the classifier, and propose hyperspherical consistency regularization (HCR) to constrain the latent hyperspherical space. As shown in Figure 2, a decent classifier is able to find an optimal hyperplane in a hypersphere manifold, and data points on the classifier’s hyperplane can be reprojected on a hypersphere. HCR assumes data points on the projection head’s hypersphere and the classifier’s hypersphere have similar geometric structures, and preserves such structures by making distributions of pairwise distances consistent. Experiments on semi-supervised learning and weakly-supervised learning indicate HCR can considerably improve the generalization ability.

2. Related works

2.1. Contrastive learning

Self-supervised learning designs pretext tasks [19, 58, 61, 91] to produce supervision signals derived from the data itself, while contrastive learning is its subset that aims to group similar samples closer and diverse samples away from each other [7, 9–11, 20, 24, 27, 28, 45, 74, 80, 83, 90, 94]. Inspired by the important technique $\ell_2$ normalization on metric learning [52, 68, 77], [83] takes the class-wise supervision to the extreme of instance-wise supervision and tries to maximally scatter the features of samples over the unit hypersphere. Most of the subsequent works [7, 9–11, 20, 24, 27, 28, 45] on contrastive learning employ $\ell_2$ normalization as a standard setting, while [80] highlights $\ell_2$ normalization helps contrastive learning optimize uniformity of the induced distribution of the features on the hypersphere together with the alignment of features from positive pairs. Representation learning benefits from the desirable traits of placing features on the unit hypersphere that improving training stability and separable ability.

Extending contrastive learning to semi-supervised learning or weakly-supervised learning is straightforward. SupCon [34] proposes class-wise contrastive loss under fully-supervised setting and inspires researchers to focus on the power of contrastive learning in supervised scenarios. Self-Tuning [81] explores group contrastive learning and tackles confirmation bias and model shift issues in an efficient one-stage framework towards data-efficient transfer learning and semi-supervised learning. CoMatch [43] unifies contrastive learning, consistency regularization, entropy minimization and graph-based SSL to mitigate confirmation bias in pseudo-label-based semi-supervised learning. PSC [79] proposes a hybrid network that jointly performs both self-supervised learning and prototypical supervised contrastive learning in a cumulative learning manner. BalFeat [32] combines strengths of supervised methods and contrastive methods to learn representations that are both discriminative and balanced. MoPro [44] simultaneously optimizes classical supervised loss and prototypical contrastive loss using momentum prototypes and tries to achieve robust weakly-supervised learning. Co-learning [71] rethinks that the co-training-based noisy label learning methods provide limited information gain since the differences between two networks of the same architecture mainly come from random initialization. Thus, this method explores intrinsic similarity and structural similarity to combat noisy labels.

These methods [12, 32, 43, 71, 79, 81] have similar architectures that concurrently optimize typical contrastive learning and supervised learning with certain techniques. We can see this manner from a different perspective: both label-dependent supervised learning and feature-dependent contrastive learning are pretext tasks that aim to learn proper representations. Previous works [2, 17] combine different pretext tasks to boost self-supervised learning performance and find there exist relationships between pretext tasks in which regularization is in need. Thus, our method regularizes the implicit connections between supervised learning and self-supervised learning in hypersphere space as shown in Figure 3. HCR builds a bridge between classical supervised learning and pretext tasks in self-supervised learning, and the regularization is plug-and-play to apply in these joint-learning methods.
2.2. Learning on the hypersphere

There are quite a number of methods that learn representations on the hypersphere \[8, 15, 16, 30, 46, 49–51, 53, 54, 60, 78, 87\] and show that the key semantics in neural networks is angular information instead of magnitude. MHE \[49\] draws inspiration from the Thomson problem to regularize networks with a minimum hyperspherical energy objective for improving the generalization ability of networks. CoMHE \[46\] shows that naively minimizing hyperspherical energy suffers from difficulties due to highly nonlinear and non-optimization, and proposes projecting neurons to suitable subspaces where hyperspherical energy can get minimized efficiently. Moreover, Johnson-Lindenstrauss lemma \[14\] establishes a guarantee for CoMHE’s projections. SphereGAN \[60\] remaps Euclidean feature spaces into the hypersphere by geometric transformation and calculates geometric moments for minimizing the multiple Wasserstein distances of probability measures on the hypersphere. Our work reprojects the Euclidean feature space of the classifier into a hypersphere and explores its connection with the projection head’s hypersphere.

3. Methods

3.1. Preliminaries

HCR focuses on regularizing the manner that is jointly training supervised learning and self-supervised learning and tries to find their relationships. Suppose \(\mathcal{X} \subset \mathbb{R}^n\) is the \(n\)-dimensional Euclidean image space, and \(\mathcal{Y} = \{0, 1\}^c\) is the ground-truth label space with \(c\) classes in an one-hot manner if the label exists. The usual framework consists of a classifier \(g : \mathbb{R}^{D_f} \rightarrow \mathbb{R}^{D_g}\) and a projection head \(h : \mathbb{R}^{D_f} \rightarrow \mathbb{S}^{D_h-1}\) with their shared feature encoder \(f : x \in \mathcal{X} \rightarrow \mathbb{R}^{D_f}\), where \(D_f, D_g, D_h\) denote the dimension of output Euclidean spaces from \(f, g, h\) respectively. HCR imposes constraints in hyperspherical spaces so that the classifier \(g : \mathbb{R}^{D_f} \rightarrow \mathbb{S}^{D_g-1}\) outputs a \((D_g - 1)\)-dimensional hypersphere \(\mathbb{S}^{D_g-1}\) by mapping the original outputs to \(\ell_2\) normalized feature vectors of dimension \(D_g\).

3.2. Hyperspherical consistency regularization

As the classifier \(g(\cdot)\) and the projection head \(h(\cdot)\) perform different tasks according to the same features from the feature encoder \(f(\cdot)\), HCR assumes there exists a distance-preserving mapping \(F : \mathbb{R}^{D_h} \rightarrow \mathbb{R}^{D_h}\) and its inverse mapping \(F^{-1} : \mathbb{R}^{D_h} \rightarrow \mathbb{R}^{D_h}\) that establish the connections of points on the hyperspheres. We argue that the relationship of the hyperspheres from different tasks can be characterized by the geometric property. Here, we consider the pairwise distance as the key geometry property, and force points on the classifier’s hypersphere to have a similar structure as the projection head’s, as shown in Figure 4.

![Figure 4. Preserve the geometry structure of data points lies on hyperspheres by their pairwise distances.](image)

We first define the pairwise distances on the respective hyperspheres as:

\[
\begin{align*}
    d_g(x_i, x_j) &= \|g \circ f(x_i) - g \circ f(x_j)\|, \\
    d_h(x_i, x_j) &= \|h \circ f(x_i) - h \circ f(x_j)\|,
\end{align*}
\]

(1)

where \(\| \cdot \|\) denotes Euclidean distance. While \(x_i, x_j \in \mathcal{X}\) and \(i \neq j\), we use \(d_g\) and \(d_h\) to represent the sets of pairwise distances on \(\mathcal{X}\) for notation simplicity. To measure the pairwise distances on hyperspheres with different dimensions, we define similarity metrics \(p(d_g)\) and \(q(d_h)\) that are considered to be normal distributions multiplied by constant terms (see Sec 4.1):
projections presented as:

In such a setting, the entire objective function can be represented learning and self-supervised learning simultaneously. HCR implicitly makes agreements on latent hyperspherical spaces between contrastive learning and supervised learning. Moreover, SupCon directly imposes contrastive loss to standard cross entropy, which requires a full exploration of label information so that it is limited in supervised learning fashion. HCR forces supervised learning to imitate contrastive learning in latent spaces without extra label information so that it can conveniently fit into semi-supervised learning and weakly-supervised learning frameworks.

4. Theoretical Insights

This section is inspired by rigorous theoretical results from [1, 3, 6, 13, 14, 18, 21, 31, 35, 39, 55, 75] and provides theoretical and intuitive perspectives about HCR.

4.1. Distribution of distance on a hypersphere

Theorem 1. (Asymptotic form of the distribution of Euclidean distance in a hypersphere for large dimensions). Given the $D$-dimensional hypersphere $S^D(a)$ with radius $a$, $x_i$ and $x_j (\forall i \neq j)$ are any two points chosen at random in $S^D(a)$ whose Euclidean distance is denoted by $r (0 \leq r \leq 2a)$. Then, the asymptotic distribution of $r$ is $N(\sqrt{2a}, \frac{a^2}{2D})$ as $D \to \infty$.

Theorem 1, which has been heavily studied in [1, 21, 55], tells us the distribution of Euclidean distance in a hypersphere obeys normal distribution as the dimension of the hypersphere becomes large. Though HCR only considers the case that points on the hypersphere surface and ignores the points inside, it still agrees with this theorem. HCR models the pairwise distance distribution as normal distribution and tries to utilize pairwise distances as a key property for preserving the geometric structure of the projection head’s outputs. Thus, HCR builds a bridge between feature-dependent and label-dependent information.

4.2. Connections between hyperspheres

Theorem 2. (Johnson-Lindenstrauss lemma). Let $\epsilon \in (0, 1)$. Let $N, D_g \in \mathbb{N}$ such that $D_g \leq C \epsilon^{-2} \log N$, for a large enough absolute constant $C$. Let $H \subseteq \mathbb{R}^{D_h}$ be a set of $N$ points. There exists a linear mapping $F : \mathbb{R}^{D_h} \to \mathbb{R}^{D_f}$, such that for all $h_i, h_j \in H$:

$$\epsilon ||h_i - h_j||^2 \leq |F \circ h_i - F \circ h_j|^2 \leq (1 + \epsilon) ||h_i - h_j||^2.$$  

(5)

The famous Johnson-Lindenstrauss lemma has found numerous applications that includes searching for graph embedding, manifold learning and dimension reduction. Here, this lemma guarantees the projection of two points from high-dimensional space to low-dimensional space preserves their Euclidean distance with high probability. Though the dimensions of feature projections $h \circ f(x)$ and logits $g \circ f(x)$ are usually not the same, HCR preserves their relative distances under this theorem.

\[
p(d_g) = C_g \frac{1}{\sigma_g \sqrt{2\pi}} \exp \left[ \frac{-1}{2} \frac{(d_g - \mu_g)^2}{\sigma_g^2} \right],
\]

\[
q(d_h) = C_h \frac{1}{\sigma_h \sqrt{2\pi}} \exp \left[ \frac{-1}{2} \frac{(d_h - \mu_h)^2}{\sigma_h^2} \right],
\]  

(2)

where $C_g, C_h$ are constants that forces the similarity metric to be in $[0, 1]$. $\sigma_g, \mu_g, \sigma_h, \mu_h$ can be chosen according to the situations. For the convenience of optimization, we empirically assume $p(d_g), q(d_h) \sim N(0, \frac{1}{2})$ in all experiments except especially mentioned.

As contrastive learning tries to push away samples from different classes and pull together samples from the same classes, the respective similarity metrics are supposed to be approaching either zero or one. Thus, we define the objective of HCR that minimizing the binary cross entropy (BCE) between $p(d_g)$ and $q(d_h)$:

\[
HCR(p(d_g), q(d_h)) = BCE(p(d_g) \mid \mid q(d_h)) = \log (1 - p(d_g)) - (1 - p(d_g)) \log (1 - q(d_h)),
\]

(3)

Through minimizing Equation 3, the mutual information $I(g \circ f(x), h \circ f(x))$ between logits $g \circ f(x)$ and feature projections $h \circ f(x)$ is implicitly maximized (see Sec 4.2).

3.3. HCR as a regularization for learning

Now that we have introduced the formulation of HCR, here we propose HCR as a regularization for semi-supervised or weakly-supervised learning. While HCR imposes the consistency between the classifier and the projection head on the hyperspherical latent spaces, it is suitable for the jointly learning manner that performs superset learning for the jointly learning manner that performs super-

4.3. Relations to supervised contrastive learning

Similar to SupCon [34], HCR leverages contrastive learning to benefit vanilla supervised learning. While SupCon explicitly pushes apart clusters of samples from different classes and pulls together clusters of samples belonging to the same class in a self-supervised contrastive manner,
Theorem 3. (Mutual information’s invariance property to reparametrization of the marginal variables). If \( H' = F(H) \) and \( G' = T(G) \) are homeomorphisms (i.e. smooth uniquely invertible maps), then the mutual information \( I(H, G) = I(H', G') \).

This theorem [13,35,75] reveals the possible connections between \( g \circ f(x) \) and \( h \circ f(x) \). We discuss that Equation 3 is preserving pairwise distances between the projection head’s and the classifier’s hyperspherical spaces. For those limited data points, we consider the mapping \( F \) from the projection head \( h \) to the classifier \( g \) is approaching bijective through preserving pairwise distances. Since when \( F \) is invertible, the mutual information is:

\[
I(h \circ f(x), g \circ f(x)) = I(h \circ f(x), F \circ (h \circ f(x))) = I(h \circ f(x), h \circ f(x))
\]

so that it is maximized. Thus, HCR preserves the distributions of pairwise distances that implicitly maximizes the mutual information \( I(h \circ f(x), g \circ f(x)) \).

5. Experiments

To validate the effectiveness of our proposed HCR, we conduct experiments on various tasks, such as semi-supervised learning, fine-grained classification, and noisy label learning, among which the latter two tasks belong to weakly-supervised learning.

5.1. Baselines

We take the recent typical works Self-Tuning [81] and Co-learning [71] as our baselines on account of their jointly learning manner, that is, both of them build the network architecture using a shared feature encoder with a classifier and a projector head and train two heads simultaneously though in different ways.

Self-Tuning unifies the exploration of labeled data and unlabeled data and the transfer of a pretrained model in a pseudo group contrast (PGC) mechanism. The vanilla contrastive learning maximizes the similarity between query \( g \) with its corresponding positive key \( k_0 \) (a different view of the same data sample):

\[
\mathcal{L}_{CL} = - \log \frac{\exp(q \cdot k_0 / \tau)}{\exp(q \cdot k_0 / \tau) + \sum_{d=1}^{D} \exp(q \cdot k_d / \tau)},
\]

where \( \tau \) is a hyperparameter for temperature scaling. Self-Tuning modifies the contrastive mechanism by introducing a group of positive keys from samples with the same pseudo label to contrast with other samples as follows:

\[
\mathcal{L}_{PGC} = - \frac{1}{D + 1} \sum_{d=0}^{D} \log \frac{\exp(q \cdot k_0^y / \tau)}{\exp(q \cdot y^y / \tau) + \text{Neg}},
\]

where \( \text{Neg} = \sum_{c=1}^{(1,2,...,C)} \sum_{i=1}^{D} \exp(q \cdot k_i^c / \tau) \), \( D \) is the number of classes, \( \hat{y} \) denotes the pseudo label. Equation 7 and Equation 8 are corresponding to the unlabeled loss \( \mathcal{L}_{u} \) in Equation 4.

Self-Tuning has bridged supervised learning and self-supervised learning by guiding the contrastive mechanism of the projection head through pseudo labels produced by the classifier. However, this scheme can only help the feature encoder obtain decent representations while the bias of the classifier remains unavoidable. Thus, we expect the projection head to benefit the classifier as well through HCR as shown in Figure 5.

Co-learning is a recent work that challenges the co-training scheme in noisy label learning. It provides perspectives from both supervised learning and self-supervised learning through the above mentioned jointly learning manner. As the classifier is extremely unreliable in noisy learning settings, Co-learning also proposes a structural similarity that imposes structure-preserving constraints similar to HCR. Co-learning directly assumes the pairwise distance follows a normal distribution and minimizes the Kullback–Leibler (KL) divergence of the distributions between the projection head and the classifier.

However, Co-learning cannot theoretically guarantees that the structure-preserving constraints are well performed. As shown in Figure 6, we train Co-learning and HCR on the CIFAR-10 dataset with 80% symmetric noise for a hundred epochs, and present the hyperspherical distance distributions. The initial hyperspherical pairwise distances of the projection head are roughly in \([0, 0.5]\). After training, they become large, which suggests the conclusion of [80] is correct, i.e., contrastive learning is trying to make features uniformly distributed on the hypersphere rather than concentrating in a local area. Moreover, the distance distribution learned by HCR preserves the structure better than Co-learning’s because the classifier’s distance distribution is balanced referring to the projection head’s.
5.2. Semi-supervised learning

Following the same experimental setting as [81], we compare Self-Tuning with HCR against three classical semi-supervised learning methods: Pseudo-Labeling [41], II-model [40], and Mean Teacher [73], as well as three recent methods UDA [85], FixMatch [69], SimCLRv2 [10], and Self-Tuning itself [81]. Experiments are conducted on three mainstream visual datasets: Stanford Cars [36], FGVC Aircraft [56], CUB-200-2011 [76], and CIFAR-100 [37]. Stanford Cars contains 16185 images of 196 classes of cars, and the pixel resolution is 360 × 240. FGVC Aircraft consists of 10000 images of 100 different aircraft model variants. The image resolution is about 1-2M pixels, but its width and height are not fixed. CUB-200-2011 is a dataset with totally 6033 images of 200 bird species, and each image has about less than 250 thousand pixels. CIFAR-100 is a classical visual dataset with 100 classes and 600 images per class, and the image resolution is 32 × 32.

For a fair comparison, all these methods implement a ResNet-50 model and initialize it from ImageNet-pretrained weights. Besides, we remove the last layer of the pretrained model and add the projection head h and classifier g with randomly initialized weights. The default temperature τ is 0.07, and the learning rate is 0.001. The optimizer follows the original Self-Tuning, which is SGD with a momentum of 0.9. Experiments are repeated three times with different random seeds, and we report the average test accuracy of three trials for each experiment. When we reproduce Self-Tuning, we unexpectedly find the results are better than its paper reported, thus we honestly report the reproduced results rather than using results from its paper.

As reported in Table 1, HCR significantly improves the performance of Self-Tuning by an average of 2.30% in different label proportions. Moreover, HCR obtains averagely 3.77% improvements under the condition of 15% label proportion, which indicates the effectiveness of HCR with extremely few labels.

Table 2 shows results on FGVC Aircraft dataset. As we can see, the observations are consistently the same as those for Stanford Cars dataset, which is, HCR is still able to obtain large gains (averagely 3.03%) even Self-Tuning has achieved a very high accuracy. Also, the lower the label proportion, the larger the improvements brought by HCR.

<table>
<thead>
<tr>
<th>Method</th>
<th>Label Proportion</th>
<th>15%</th>
<th>30%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudo-Labeling</td>
<td></td>
<td>40.93±0.23</td>
<td>67.02±0.19</td>
<td>78.71±0.30</td>
</tr>
<tr>
<td>II-model</td>
<td></td>
<td>45.19±0.21</td>
<td>57.29±0.26</td>
<td>64.18±0.29</td>
</tr>
<tr>
<td>Mean Teacher</td>
<td></td>
<td>54.28±0.14</td>
<td>66.02±0.21</td>
<td>74.24±0.23</td>
</tr>
<tr>
<td>UDA</td>
<td></td>
<td>39.90±0.43</td>
<td>64.16±0.40</td>
<td>71.86±0.56</td>
</tr>
<tr>
<td>FixMatch</td>
<td></td>
<td>49.86±0.27</td>
<td>77.54±0.29</td>
<td>84.78±0.33</td>
</tr>
<tr>
<td>SimCLRv2</td>
<td></td>
<td>45.74±0.16</td>
<td>61.70±0.18</td>
<td>77.49±0.24</td>
</tr>
</tbody>
</table>

Table 1. Classification accuracy (%) ↑ of semi-supervised learning methods on Stanford Cars dataset (ResNet-50 pretrained).

Table 2. Classification accuracy (%) ↑ of semi-supervised learning methods on FGVC Aircraft dataset (ResNet-50 pretrained).

<table>
<thead>
<tr>
<th>Method</th>
<th>Label Proportion</th>
<th>15%</th>
<th>30%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudo-Labeling</td>
<td></td>
<td>46.83±0.30</td>
<td>62.77±0.31</td>
<td>73.21±0.39</td>
</tr>
<tr>
<td>II-model</td>
<td></td>
<td>37.72±0.25</td>
<td>58.49±0.26</td>
<td>65.63±0.36</td>
</tr>
<tr>
<td>Mean Teacher</td>
<td></td>
<td>51.59±0.23</td>
<td>71.62±0.29</td>
<td>80.31±0.32</td>
</tr>
<tr>
<td>UDA</td>
<td></td>
<td>43.96±0.45</td>
<td>64.17±0.49</td>
<td>67.42±0.53</td>
</tr>
<tr>
<td>FixMatch</td>
<td></td>
<td>55.53±0.26</td>
<td>71.35±0.35</td>
<td>78.34±0.43</td>
</tr>
<tr>
<td>SimCLRv2</td>
<td></td>
<td>40.78±0.21</td>
<td>59.03±0.29</td>
<td>68.54±0.30</td>
</tr>
</tbody>
</table>

Table 2. Classification accuracy (%) ↑ of semi-supervised learning methods on FGVC Aircraft dataset (ResNet-50 pretrained).

We report results on CUB-200-2011 dataset in Table 3. It can be seen that applying HCR yields better performance than the original Self-Tuning.

Note that the improvement on CUB-200-2011 dataset is slightly less than the former two datasets. The reason is that the average number of samples per class of CUB-200-2011 is much less than Stanford Cars and FGVC Aircraft.
datasets. It’s difficult for HCR to capture the structure of data while those data are sparsely distributed on the hypersphere. We analyze the relationship between average samples per class and the improvements as shown in Figure 7.

Except for the visual datasets evaluated in Self-Tuning, we perform experiments on the standard semi-supervised learning benchmark CIFAR-100 dataset. The results are shown in Table 4 and Table 5. Self-Tuning implements EfficientNet-B2 model [72] as the pretrained weights of WRN-28-8 are not available. While FixMatch obtains a higher error rate with EfficientNet-B2 than WRN-28-8, Self-Tuning outperforms those methods on WRN-28-8, HCR here further increases its leading.

Besides, under extremely few labels condition, HCR strongly outperforms other methods through promoting Self-Tuning by 4.46% which is definitely a large margin. We believe our proposed HCR can play a significant role in bridging feature-dependent and label-dependent information, especially in the case of few labels being available.

5.3. Fine-grained classification

We conduct experiments on fine-grained classification using fully-labeled Stanford Cars, FGVC Aircraft, and CUB-200-2011 datasets. The results in Table 6 show that HCR performs consistently better than the baseline. FGVC Aircraft still gets the most improvements, benefited from its large average number of samples per class as we have mentioned above. We believe HCR can not only help semi-supervised conditions but also difficult supervised learning.

5.4. Noisy label learning

We follows the same experimental settings as [71] to compare Co-learning with HCR against other co-training-based noisy label learning methods: Decoupling [57], Co-
Co-learning [22], Co-teaching+ [88], JoCoR [82], and Co-learning itself [71]. We conduct experiments on CIFAR-100 dataset with three different types of noise, i.e., symmetric, asymmetric, and instance-dependent. The details of these noise types are in Appendix. Among these noisy types, we recognize instance-dependent (or feature-dependent) noise as a more realistic setting because human annotations are prone to different levels of errors for tasks with varying difficulty levels. Following [71], we report the average test accuracy over the last 10 epochs of five trials for each experiment. The base model is ResNet-18.

Table 7 shows the results on symmetric noise. As it is the simplest synthetic noise type, we perform experiments on high noise ratios as 50% and 80%. While Co-learning has already shown amazing results on high noise ratios, HCR further improves Co-learning by averagely 3.92% under different noise ratios.

We report the results on asymmetric noise in Table 8. HCR significantly improves Co-learning by 3.74% on average. Moreover, HCR presents more stable results than Co-learning as the standard deviations are minor.

When it comes to instance-dependent noise, HCR assists Co-learning in obtaining 1.52% gains averagely as shown in Table 9. As Co-learning has utilized feature-dependent information, HCR’s improvements on instance-dependent noise are slightly less than the former two noise types.

We also conduct experiments on CIFAR10 with extremely high noise and show the results in Figure 8. While Co-learning suffers from overfitting on noisy labels, HCR consistently works well and performs great robustness.

<table>
<thead>
<tr>
<th>Method</th>
<th>sym-20%</th>
<th>sym-50%</th>
<th>sym-80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standrad CE</td>
<td>57.79±0.44</td>
<td>33.75±0.46</td>
<td>8.64±0.22</td>
</tr>
<tr>
<td>Decoupling</td>
<td>56.18±0.32</td>
<td>31.58±0.54</td>
<td>7.71±0.23</td>
</tr>
<tr>
<td>Co-teaching</td>
<td>64.28±0.32</td>
<td>32.62±0.51</td>
<td>6.65±0.71</td>
</tr>
<tr>
<td>Co-teaching+</td>
<td>55.40±0.71</td>
<td>26.49±0.45</td>
<td>8.57±1.55</td>
</tr>
<tr>
<td>JoCoR</td>
<td>62.29±0.71</td>
<td>30.19±0.60</td>
<td>6.84±0.92</td>
</tr>
<tr>
<td>Co-learning</td>
<td>66.58±0.15</td>
<td>55.54±0.43</td>
<td>35.45±0.79</td>
</tr>
<tr>
<td>Co-learning+HCR</td>
<td>70.27±0.32</td>
<td>59.93±0.25</td>
<td>39.14±0.47</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>ins-20%</th>
<th>ins-30%</th>
<th>ins-40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standrad CE</td>
<td>55.45±0.54</td>
<td>48.77±0.47</td>
<td>41.30±0.27</td>
</tr>
<tr>
<td>Decoupling</td>
<td>52.20±0.48</td>
<td>45.32±0.83</td>
<td>36.33±0.47</td>
</tr>
<tr>
<td>Co-teaching</td>
<td>55.16±0.61</td>
<td>45.24±0.37</td>
<td>34.64±1.00</td>
</tr>
<tr>
<td>Co-teaching+</td>
<td>50.37±0.85</td>
<td>40.73±0.58</td>
<td>32.15±0.80</td>
</tr>
<tr>
<td>JoCoR</td>
<td>54.21±0.34</td>
<td>45.03±0.52</td>
<td>34.08±1.05</td>
</tr>
<tr>
<td>Co-learning</td>
<td>69.42±0.42</td>
<td>65.45±0.86</td>
<td>60.40±1.37</td>
</tr>
<tr>
<td>Co-learning+HCR</td>
<td>70.03±0.31</td>
<td>66.89±0.41</td>
<td>62.91±0.84</td>
</tr>
</tbody>
</table>

6. Conclusion

In this paper, we borrow several theoretical insights from geometry and propose a novel consistency regularization method for semi-supervised and weak-supervised learning, called hyperspherical consistency regularization (HCR), to encourage the pairwise distance distribution of the classifier to be similar to the distribution of the projection head in the latent space. HCR can be conveniently implemented to those jointly learning methods as a plug-in regularization, or be applied to a vanilla supervised learning network with only an additional projection head. Through extensive experiments on semi-supervised learning, fine-grained classification and noisy label learning, HCR shows consistent improvements on these tasks. In general, HCR cast a novel view on leveraging self-supervised learning to assist data-efficient and robust deep learning by introducing hyperspherical consistency.

Acknowledgement This work is supported in part by the Science and Technology Innovation 2030 - Major Project (No. 2021ZD0150100) and National Natural Science Foundation of China (No. U21A20427).
References


[56] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classi-


[82] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with


