GENERALIZING AND DECOUPLING NEURAL COLLAPSE VIA HYPERSPHERICAL UNIFORMITY GAP

Anonymous authors
Paper under double-blind review

ABSTRACT

The neural collapse (NC) phenomenon describes an underlying geometric symmetry for deep neural networks, where both deeply learned features and classifiers converge to a simplex equiangular tight frame. It has been shown that both cross-entropy loss and mean square error can provably lead to NC. Inspired by how NC characterizes the training target of neural networks, we decouple NC into two objectives: minimal intra-class variability and maximal inter-class separability. We then introduce the concept of hyperspherical uniformity (which characterizes the degree of uniformity on the unit hypersphere) as a unified framework to quantify these two objectives. Finally, we propose a generic objective – hyperspherical uniformity gap (HUG), which is defined by the difference between inter-class and intra-class hyperspherical uniformity. HUG not only provably converges to NC, but also decouples NC into two separate objectives. Unlike cross-entropy loss that couples intra-class compactness and inter-class separability, HUG enjoys more flexibility and serves as a good alternative loss function. Empirical results show that HUG works well in terms of generalization, calibration and robustness.

1 INTRODUCTION

Recent years have witnessed the great success of deep representation learning in a variety of applications ranging from computer vision [28], natural language processing [12] to game playing [44, 50]. Despite such a success, how deep representations can generalize to unseen scenarios and when they might fail remain a black box. Deep representations are typically learned by a multi-layer network with cross-entropy (CE) loss optimized by stochastic gradient descent. In this simple setup, [63] has shown that zero loss can be achieved even with arbitrary label assignment. After continuing to train the neural network past zero loss with CE, [47] discovers an intriguing phenomenon called neural collapse (NC). NC can be summarized as the following characteristics:

- **Intra-class variability collapse**: Intra-class variability of last-layer features collapses to zero, indicating that all the features of the same class concentrate to their intra-class feature mean.
- **Convergence to simplex ETF**: After being centered at their global mean, the class-means are both linearly separable and maximally distant on a hypersphere. Formally, the class-means form a simplex equiangular tight frame (ETF) which is a symmetric structure defined by a set of maximally distant and pair-wise equiangular points on a hypersphere.
- **Convergence to self-duality**: The linear classifiers, which live in the dual vector space to that of the class-means, converge to their corresponding class-mean and also form a simplex ETF.
- **Nearest decision rule**: The linear classifiers behave like nearest class-mean classifiers.

The NC phenomenon suggests two general principles for deeply learned features and classifiers: minimal intra-class compactness of features (*i.e.*, features of the same class collapse to a single point), and maximal inter-class separability of classifiers / feature mean (*i.e.*, classifiers of different classes have maximal angular margins). While these two principles are largely independent, popular loss functions such as CE and square error (MSE) completely couple these two principles together. Since there is no trivial way for CE and MSE to decouple these two principles, we identify a novel quantity – hyperspherical uniformity gap (HUG), which not only characterizes intra-class feature compactness and inter-class classifier separability as a whole, but also fully decouples these two principles. The decoupling enables HUG to separately model intra-class compactness and inter-class

separability, making it highly flexible. More importantly, HUG can be directly optimized and used to train neural networks, serving as an alternative loss function in place of CE and MSE for classification. HUG is formulated as the difference between inter-class and intra-class hyperspherical uniformity. Hyperspherical uniformity [38] quantifies the uniformity of a set of vectors on a hypersphere and is used to capture how diverse these vectors are on a hypersphere. Thanks to the flexibility of HUG, we are able to use many different formulations to characterize hyperspherical uniformity, including (but not limited to) minimum hyperspherical energy (MHE) [35], maximum hyperspherical separation (MHS) [38] and maximum gram determinant (MGD) [38]. Different formulations yield different interpretation and optimization difficulty (*e.g.*, HUG with MHE is easy to optimize, HUG with MGD has interesting connection to geometric volume), thus leading to different performance.

Similar to CE loss, HUG also provably leads to NC under the setting of unconstrained features [42]. Going beyond NC, we hypothesize a generalized NC (GNC) that extends the original NC to the scenario where there is no constraint for the number of classes and the feature dimension. NC requires the feature dimension no smaller than the number of classes while GNC no longer requires this. We further prove that HUG also leads to GNC at its objective minimum.

Another motivation behind HUG comes from the classic Fisher discriminant analysis (FDA) [14] where the basic idea is to find a projection matrix $\boldsymbol{T}$ that maximizes between-class variance and minimizes within-class variance. What if we directly optimize the input data (without any projection) rather than optimizing the linear projection in FDA? We make a simple derivation below:

$$\textbf{Projection FDA:} \quad \max_{\boldsymbol{T}\in\mathbb{R}^{d\times r}} \operatorname{tr}\left(\left(\boldsymbol{T}^\top \boldsymbol{S}_w \boldsymbol{T}\right)^{-1} \boldsymbol{T}^\top \boldsymbol{S}_b \boldsymbol{T}\right) \qquad \textbf{Data FDA:} \quad \max_{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n\in\mathbb{S}^{d-1}} \operatorname{tr}\left(\boldsymbol{S}_b\right) - \operatorname{tr}\left(\boldsymbol{S}_w\right)$$

where the between-class scatter matrix is $\boldsymbol{S}_w = \sum_{i=1}^{C} \sum_{j\in A_c} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top$, the within-class scatter matrix is $\boldsymbol{S}_b = \sum_{i=1}^{C} n_i (\boldsymbol{\mu}_i - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_i - \bar{\boldsymbol{\mu}})^\top$, n_i is the number of samples in the i-th class, n is the total number of samples, $\boldsymbol{\mu}_i = n_i^{-1} \sum_{j\in A_c} \boldsymbol{x}_j$ is the i-th class-mean, and $\bar{\boldsymbol{\mu}} = n^{-1} \sum_{j=1}^{n} \boldsymbol{x}_j$ is the global mean. By considering class-balanced data on the unit hypersphere, optimizing data FDA is equivalent to simultaneously maximizing $\operatorname{tr}(\boldsymbol{S}_b)$ and minimizing $\operatorname{tr}(\boldsymbol{S}_w)$. Maximizing $\operatorname{tr}(\boldsymbol{S}_b)$ encourages inter-class separability and is a necessary condition for hyperspherical uniformity.[1] Minimizing $\operatorname{tr}(\boldsymbol{S}_w)$ encourages intra-class feature collapse, reducing intra-class variability. Therefore, HUG can be viewed a generalized FDA criterion for learning maximally discriminative features.

However, one may ask the following questions: *Why is HUG useful if we already have the FDA criterion? Could we simply optimize data FDA?* In fact, the FDA criterion has many degenerate solutions. For example, we consider a scenario of 10-class balanced data where all features from the first 5 classes collapse to the north pole on the unit hypersphere and features from the rest 5 classes collapse to the south pole on the unit hypersphere. In this case, $\operatorname{tr}(\boldsymbol{S}_w)$ is already minimized since it achieves the minimum zero. $\operatorname{tr}(\boldsymbol{S}_b)$ also achieves its maximum n at the same time. In contrast, HUG naturally generalizes FDA without having these degenerate solutions and serves as a more reliable criterion for training neural networks. We summarize our contributions below:

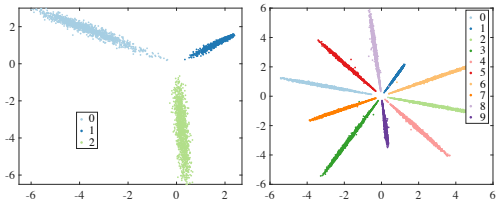
- We decouple the NC phenomenon into two separate learning objectives: maximal inter-class separability (*i.e.*, maximally distant class feature mean and classifiers on the hypersphere) and minimal intra-class variability (*i.e.*, intra-class features collapse to a single point on the hypersphere).
- Based on the two principled objectives induced by NC, we hypothesize the generalized NC which generalizes NC by dropping the constraint on the feature dimension and the number of classes.
- We identify a general quantity called hyperspherical uniformity gap, which well characterizes both inter-class separability and intra-class variability. Different from the widely used CE loss, HUG naturally decouples both principles and thus enjoys better modeling flexibility.
- Under the HUG framework, we consider three different choices for characterizing hyperspherical uniformity: minimum hyperspherical energy, maximum hyperspherical separation and maximum Gram determinant. There also exist many other choices that may work equally well. HUG provides a unified framework for different characterizations of hyperspherical uniformity to supervise neural networks. We give both theoretical and empirical insights to validate HUG's effectiveness.

[1] We first obtain the upper bound n of $\operatorname{tr}(\boldsymbol{S}_b)$ from $\operatorname{tr}(\boldsymbol{S}_b) = \sum_{i=1}^{C} n_i \|\boldsymbol{\mu}_i - \boldsymbol{\mu}\|_F^2 \leq \sum_{i=1}^{C} n_i \|\boldsymbol{\mu}_i\| \cdot \|\boldsymbol{\mu}\| \leq n$. Because a set of vectors $\{\boldsymbol{\mu}_i\}_{i=1}^{n}$ achieving hyperspherical uniformity has $\mathbb{E}_{\boldsymbol{\mu}_1,\cdots,\boldsymbol{\mu}_n}\{\|\bar{\boldsymbol{\mu}}\|\} \to \mathbf{0}$ (as n grows larger) [15]. Then we have that $\operatorname{tr}(\boldsymbol{S}_b)$ attains n. Therefore, vectors achieving hyperspherical uniformity are one of its maximizers. $\operatorname{tr}(\boldsymbol{S}_w)$ can simultaneously attain its minimum if intra-class features collapse to a single point.

2 On Generalizing and Decoupling Neural Collapse

NC describes an intriguing phenomenon for the distribution of last-layer features and classifiers in overly-trained neural networks, where both features and classifiers converge to ETF. However, ETF can only exist when the feature dimension d and the number of classes C satisfy $d \geq C-1$. This is not always true for deep neural networks. For example, neural networks for face recognition are usually trained by classifying large number of classes (*e.g.*, more than 85K classes in [17]), and the feature dimension (*e.g.*, 512 in SphereFace [34]) is usually much less than the number of classes. In general, when the number of classes is already large, it is infeasible to use a larger feature dimension. Thus a natural question arises: *what will happen in this case if a neural network is fully trained?*

Motivated by this question, we conduct a simple experiment to simulate the case of $d \geq C-1$ and the case of $d < C-1$. Specifically, we train a convolutional neural network (CNN) on MNIST with feature dimension 2. For the case of $d \geq C-1$, we use only 3 classes (digit 0,1,2) as the training set. For the case of $d < C-1$, we use all 10 classes as the training set. We visualize the learned features of both cases in Figure 1. The results verify the case of $d \geq C-1$ indeed approaches to NC, and ETF does not exist in the case of $d < C-1$. Interestingly, one can observe that learned features in both cases approach to the configuration of equally spaced frames on the hypersphere. To accommodate the case of $d < C-1$, we extend NC to the generalized NC by hypothesizing that last-layer inter-class features and classifiers tend to converge to equally spaced points on the hypersphere, which can be characterized by hyperspherical uniformity.



(a) 2D feature with 3 classes (b) 2D feature with 10 classes

Figure 1: 2D learned feature visualization on MNIST. The features are inherently 2-dimensional and are plotted without visualization tools. (a) Case 1: $d=2, C=3$; (b) Case 2: $d=2, C=10$.

Generalized Neural Collapse (GNC)

We define the feature global mean as $\boldsymbol{\mu}_G = \text{Ave}_{i,c}\boldsymbol{x}_{i,c}$ where $\boldsymbol{x}_{i,c} \in \mathbb{R}^d$ is the last-layer feature of the i-th sample in the c-th class, the feature class-mean as $\boldsymbol{\mu}_c = \text{Ave}_i \boldsymbol{x}_{i,c}$ for different classes $c \in \{1, \cdots, C\}$, the feature within-class covariance as $\boldsymbol{\Sigma}_W = \text{Ave}_{i,c}(\boldsymbol{x}_{i,c}-\boldsymbol{\mu}_c)(\boldsymbol{x}_{i,c}-\boldsymbol{\mu}_c)^\top$ and the feature between-class covariance as $\boldsymbol{\Sigma}_B = \text{Ave}_c(\boldsymbol{\mu}_c-\boldsymbol{\mu}_G)(\boldsymbol{\mu}_c-\boldsymbol{\mu}_G)^\top$. GNC states that

- **(1) Intra-class variability collapse**: Intra-class variability of last-layer features collapse to zero, indicating that all the features of the same class converge to their intra-class feature mean. Formally, GNC has that $\boldsymbol{\Sigma}_B^\dagger \boldsymbol{\Sigma}_W \to \mathbf{0}$ where $\dagger$ denotes the Moore-Penrose pseudoinverse.
- **(2) Convergence to hyperspherical uniformity**: After being centered at their global mean, the class-means are both linearly separable and maximally distant on a hypersphere. Formally, the class-means converge to equally spaced points on a hypersphere, *i.e.*,

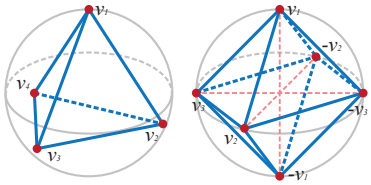
$$\sum_{c\neq c'} \|\hat{\boldsymbol{\mu}}_c - \hat{\boldsymbol{\mu}}_{c'}\|^{-2} \to \min_{\hat{\boldsymbol{\mu}}_1,\cdots,\hat{\boldsymbol{\mu}}_C} \sum_{c\neq c'} \|\hat{\boldsymbol{\mu}}_c - \hat{\boldsymbol{\mu}}_{c'}\|^{-2}, \quad \|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\| - \|\boldsymbol{\mu}_{c'} - \boldsymbol{\mu}_G\| \to 0, \ \forall c \neq c' \quad (1)$$

  where $\hat{\boldsymbol{\mu}}_i = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_G\|^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_G)$. $E = \sum_{c\neq c'} \|\hat{\boldsymbol{\mu}}_c - \hat{\boldsymbol{\mu}}_{c'}\|^{-2}$ is a variational characterization of hyperspherical uniformity using Coulomb potentials (defined as *hyperspherical energy* [35]).
- **(3) Convergence to self-duality**: The linear classifiers, which live in the dual vector space to that of the class-means, converge to their corresponding class-means, leading to hyperspherical uniformity. Formally, GNC has that $\|\boldsymbol{w}_c\|^{-1}\boldsymbol{w}_c - \hat{\boldsymbol{\mu}}_c \to 0$ where $\boldsymbol{w}_c \in \mathbb{R}^d$ is the c-th classifier.
- **(4) Nearest decision rule**: The learned linear classifiers behave like the nearest class-mean classifiers. Formally, GNC has that $\arg\max_c \langle \boldsymbol{w}_c, \boldsymbol{x}\rangle + b_c \to \arg\min_c \|\boldsymbol{x} - \boldsymbol{\mu}_c\|$.

In contrast to NC, GNC further considers the case of $d < C-1$ and hypothesizes that both feature class-means and classifiers converge to hyperspherically uniform point configuration that minimizes the Coulomb potential. We also prove in Theorem 1 that GNC reduces to NC in the case of $d \geq C-1$:

Theorem 1 (Regular Simplex Optimum for GNC) *Let $f : (0, 4] \to \mathbb{R}$ be a convex and decreasing function defined at $v=0$ by $\lim_{v\to 0^+} f(v)$. If $2 \leq C \leq d+1$, then we have that the vertices of regular $(C-1)$-simplices inscribed in $\mathbb{S}^{d-1}$ with centers at the origin (equivalent to simplex ETF) minimize the hyperspherical energy $\sum_{c\neq c'} K(\hat{\boldsymbol{\mu}}_c, \hat{\boldsymbol{\mu}}_{c'})$ on the unit hypersphere $\mathbb{S}^{d-1}$ ($d \geq 3$) with the kernel as $K(\hat{\boldsymbol{\mu}}_c, \hat{\boldsymbol{\mu}}_{c'}) = f(\|\hat{\boldsymbol{\mu}}_c - \hat{\boldsymbol{\mu}}_{c'}\|^2)$. If f is strictly convex and strictly decreasing, then these are the only energy minimizing C-point configurations. Thus GNC reduces to NC when $d \geq C-1$.*

We note that Theorem 2 not only guarantees the equiangular ETF as the minimizer of the hyperspherical energy in Eq. 1, but also shows that a general family of energies also share the same minimizer (as long as the potential function f is convex and decreasing). The case of $d < C-1$ is where GNC really gets interesting but complicated. Other than the regular simplex case, we also highlight a special uniformity case of $2d = C$. In this case, we can prove in Theorem 2 that GNC(2) converges to the vertices of a cross-polytope as hyperspherical energy gets minimized. As the number of classes gets infinitely large, we show in Theorem 3 that GNC(2) leads to a point configuration that is uniformly distributed on $\mathbb{S}^{d-1}$. Additionally, we show a simple yet interesting result in Proposition 1 that the last-layer classifiers are initialized to be uniformly distributed on the hypersphere.



(a) Regular simplex (b) Cross-polytope

Figure 2: Geometric illustration in $\mathbb{R}^3$ of (a) regular simplex optimum (equivalent to simplex ETF in NC) and (b) cross-polytope optimum in GNC.

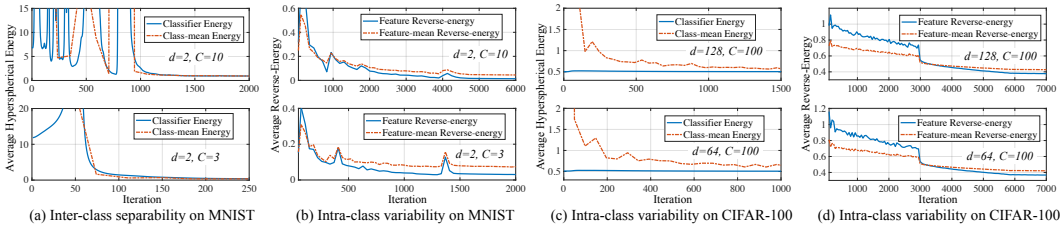
Theorem 2 (Cross-polytope Optimum for GNC) *If $C = 2d$, then the vertices of the cross-polytope are the minimizer of the hyperspherical energy in GNC(2).*

The cross-polytope optimum for GNC(2) is in fact quite intuitive, because it corresponds to the Cartesian coordinate system (up to a rotation). For example, the vertices of the unit cross-polytope in $\mathbb{R}^3$ are $(\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1)$. These 6 vectors minimize the hyperspherical energy on $\mathbb{S}^2$. We illustrate both the regular simplex and cross-polytope cases in Figure 2. For the other cases of $d < C-1$, there exists generally no simple point structure that minimizes the hyperspherical energy, as heavily studied in [10, 20, 29, 49]. For the point configurations that asymptotically minimize the hyperspherical energy as C grows larger, Theorem 3 can guarantee that these configurations asymptotically converge to a uniform distribution on the hypersphere.

Theorem 3 (Asymptotic Convergence to Hyperspherical Uniformity) *Consider a sequence of point configurations $\{\hat{\boldsymbol{\mu}}_1^C, \cdots, \hat{\boldsymbol{\mu}}_C^C\}_{C=2}^{\infty}$ that asymptotically minimizes the hyperspherical energy on $\mathbb{S}^{d-1}$ as $C \to \infty$, then $\{\hat{\boldsymbol{\mu}}_1^C, \cdots, \hat{\boldsymbol{\mu}}_C^C\}_{C=2}^{\infty}$ is uniformly distributed on the hypersphere $\mathbb{S}^{d-1}$.*

Proposition 1 (Minimum Energy Initialization) *With zero-mean Gaussian initialization (e.g., [16, 21]), the C last-layer classifiers of neural networks are initialized as a uniform distribution on the hypersphere. The expected initial energy is $C(C-1)\int_{\mathbb{S}^{d-1}}\int_{\mathbb{S}^{d-1}} \|\hat{\boldsymbol{\mu}}_c - \hat{\boldsymbol{\mu}}_{c'}\|^{-2} d\sigma_{d-1}(\hat{\boldsymbol{\mu}}_c) d\sigma_{d-1}(\hat{\boldsymbol{\mu}}_{c'})$.*

With Proposition 1, one can expect that the hyperspherical energy of the last-layer classifiers will first increase and then decrease to a lower value than the initial energy. To validate the effectiveness of our GNC hypothesis, we conduct a few experiments to show how both class feature means and classifiers converge to hyperspherical uniformity (*i.e.*, minimizing the hyperspherical energy), and how intra-class feature variability collapses to almost zero. We start with an intuitive understanding about GNC from Figure 1. The results are directly produced by the learned features without any visualization tool (such as t-SNE [57]), so the feature distribution can reflect the underlying one learned by neural networks. We observe that GNC is attained in both $d < C-1$ and $d \geq C-1$, while NC is violated in $d < C-1$ since the learned feature class-means can no longer form a simplex ETF. To see whether the same conclusion holds for higher feature dimensions, we also train two CNNs on CIFAR-100 with feature dimension as 64 and 128, respectively. The results are given in Figure 3.



(a) Inter-class separability on MNIST (b) Intra-class variability on MNIST (c) Intra-class variability on CIFAR-100 (d) Intra-class variability on CIFAR-100

Figure 3: Training dynamics of hyperspherical energy (which captures inter-class separability) and hyperspherical reverse-energy (which captures intra-class variability). (a,b) MNIST with $d = 2$, $C = 10$ and $d = 2$, $C = 3$. (c,d) CIFAR-100 with $d = 64$, $C = 100$ and $d = 128$, $C = 100$.

Figure 3 shows that GNC captures well the underlying convergence of the neural network training. Figure 3(a,c) shows that the hyperspherical energy of feature class-means and classifiers converge to a small value, verifyng the correctness of GNC(2) which indicates feature class-means converge to hyperspherical uniformity, and GNC(3) which indicates classifiers also converge to hyperspherical

uniformity. More interestingly, in the MNIST experiment, we can compute the exact minimal energy on $\mathbb{S}^1$: 2 in the case of $d=2, C=3$ ($1/3$ for average energy) and ≈ 82.5 in the case of $d=2, C=10$ (≈ 0.917 for average energy). The final average energy in Figure 3(a) matches our theoretical minimum well. From Figure 3(c), we observe that the classifier energy stays close to its minimum at the very beginning, which matches our Proposition 1 that vectors initialized with zero-mean Gaussian are uniformly distributed over the hypersphere (this phenomenon becomes more obvious in higher dimensions). To evaluate the intra-class feature variability, we consider a hyperspherical reverse-energy $E_r = \sum_{i \neq j \in A_c} \|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j\|^2$ where $\hat{\boldsymbol{x}}_i = \frac{\boldsymbol{x}_i}{\|\boldsymbol{x}_i\|}$ and A_c denotes the sample index set of the c-th class. The smaller this reverse-energy gets, the less intra-class variability it implies. We plot the reverse-energy in Figure 3(b,d) and show that the intra-class feature variability becomes smaller and smaller (approaches to zero) as GNC(1) suggests.

After establishing the GNC hypothesis, we discuss how to decouple the GNC phenomenon and how such a decoupling can enable us to propose a new objective to train neural networks. GNC(1) and GNC(2) suggest to minimize intra-class feature variability and maximize inter-class feature separability, respectively. GNC(3) and GNC(4) are in fact natural consequences if GNC(1) and GNC(2) hold. It has long been discovered [33, 53, 61] that last-layer classifiers serve as class proxies to represent the corresponding class of features, and they are also an approximation to the feature class-means. Therefore, GNC(3) is a direct consequence of GNC(1) and GNC(2). GNC(3) indicates the classifiers converge to hyperspherical uniformity, which, together with GNC(1), implies GNC(4).

Until now, it has been clear that GNC really boils down to two decoupled objectives: *maximize inter-class separability* and *minimize intra-class variability*, which again echos the goal of FDA. The problem reduces to how to effectively characterize these two objectives while being decoupled for flexibility (unlike CE or MSE). In the next section, we propose to address this problem by characterizing both objectives with a unified quantity - hyperspherical uniformity.

3 Hyperspherical Uniformity Gap

3.1 General Framework

As GNC(2) suggests, the inter-class separability is well captured by hyperspherical uniformity of feature class-means, so it is natural to directly use it as a learning target. On the other hand, GNC(1) does not suggest any easy-to-use quantity to characterize intra-class variability. We note that minimizing intra-class variability is actully equivalent to encouraging features of the same class to concentrate on a single point, which is the opposite of hyperspherical uniformity. Therefore, we can unify both intra-class variability and inter-class separability with a single characterization of hyperspherical uniformity. We propose to maximize the hyperspherical uniformity gap:

$$\max_{\{\boldsymbol{x}_i\}_{i=1}^n} \mathcal{L}_{\text{HUG}} := \alpha \cdot \underbrace{\mathcal{HU}\left(\{\hat{\boldsymbol{\mu}}_c\}_{c=1}^C\right)}_{T_b\text{: Inter-class Hyperspherical Uniformity}} - \beta \cdot \sum_{c=1}^{C} \underbrace{\mathcal{HU}\left(\{\hat{\boldsymbol{x}}_i\}_{i \in A_c}\right)}_{T_w\text{: Intra-class Hyperspherical Uniformity}} \tag{2}$$

where α, β are hyperparameters, $\hat{\boldsymbol{\mu}}_c = \frac{\boldsymbol{\mu}_c}{\|\boldsymbol{\mu}_c\|}$ is the feature class-mean projected on the unit hypersphere, $\boldsymbol{\mu}_c = \sum_{c \in A_c} \boldsymbol{x}_c$ is the feature class-mean, $\boldsymbol{x}_i$ is the last-layer feature of the i-th sample and A_c denotes the sample index set of the c-th class. $\mathcal{HU}(\{\boldsymbol{v}_i\}_{i=1}^m)$ denotes some measure of hyperspherical uniformity for vectors $\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m\}$. Eq. 2 is the general objective for HUG. Without loss of generality, we assume that the larger it gets, the stronger hyperspherical uniformity we have. We mostly focus on supervised learning with *parameteric class proxies*[2] where the CE loss is widely used as a *de facto* choice, although HUG can be used in much broader settings as discussed later. In the HUG framework, there is no longer a clear notion of classifiers (unlike the CE loss), but we still can utilize class proxies (*i.e.*, a generalized concept of classifiers) to help with the optimization.

We can observe that Eq. 2 directly optimizes the feature class-means for inter-class separability, but they are usually inefficient to compute during training (we need to compute them in every iteration). Therefore it is nontrivial to optimize the original HUG for training neural networks. A naive solution is to compute approximate feature class-mean with a few mini-batches and the gradients of inter-class hyperspherical uniformity (T_b) can be back-propagated back to the last-layer features. However, it may takes many mini-batches in order to obtain a sufficiently accurate class-mean, and

[2]Parametric class proxies are a set of parameters used to represent a group of samples in the same class. Therefore, these proxies store the information about a class. Last-layer classifiers are a typical example.

the approximation gets much more difficult with large number of classes. To address this, we employ parametric class proxies to act as representatives of intra-class features and optimize them instead of feature class-means. To make these parametric proxies become representatives of each class, we need to connect the class proxies to the features such that the gradient of T_b can have an influence on the features. To achieve this, we extend the HUG objective in Eq. 2 to cope with parametric class proxies:

$$\max_{\{\hat{\boldsymbol{x}}_i\}_{i=1}^n, \{\hat{\boldsymbol{w}}_c\}_{c=1}^C} \mathcal{L}_{\text{P-HUG}} := \alpha \cdot \underbrace{\mathcal{HU}\big(\{\hat{\boldsymbol{w}}_c\}_{c=1}^C\big)}_{\text{Inter-class Hyperspherical Uniformity}} - \beta \cdot \sum_{c=1}^C \underbrace{\mathcal{HU}\big(\{\hat{\boldsymbol{x}}_i\}_{i\in A_c}, \hat{\boldsymbol{w}}_c\big)}_{\text{Intra-class Hyperspherical Uniformity}} \tag{3}$$

where $\hat{\boldsymbol{w}}_c \in \mathbb{S}^{d-1}$ is the learnable parametric proxy for the c-th class. The intra-class hyperspherical uniformity term connects the class proxies with features by minimizing the joint hyperspherical uniformity of intra-class features and their class proxy, guiding the intra-class features to move towards their corresponding class proxy. Now the problem becomes how to update the class proxies.

Learnable proxies. As the most straightforward way, we can view the class proxy $\hat{\boldsymbol{\mu}}_i$ as a set of learnable parameters and update them with stochastic gradients, similarly to the other parameters of the neural network. In fact, learnable proxies play a role similar to the last-layer classifiers in the CE loss, and are beneficial to the optimization due to the ability of aggregating intra-class features. The only difference between learnable proxies and moving-averaged proxies is the way we update them. As GNC(3) implies, class proxies in HUG can also be used as classifiers.

Static proxies. Eq. 3 is decoupled into maximal inter-class separability and minimal intra-class variability. These two objects are independent and do not affect each other. We can thus optimize them independently. This suggests a even simpler way to assign class proxies – initializing class proxies with prespecified points that have attained hyperspherical uniformity, and fixing them in the training. There are two simple ways to obtain these class proxies: (1) minimizing their hyperspherical energy beforehand; (2) using zero-mean Gaussian to initialize the class proxies (Proposition 1 guarantees the probabilistic hyperspherical uniformity). After initialized with minimum energy, these class proxies will stay fixed and only the features are optimized towards their class proxies.

Partially learnable proxies. After the class proxies are initialized using the static way above, we can increase its flexibility by learning an orthogonal matrix for the class proxies to find a suitable orientation for them. Specifically, we can learn this orthogonal matrix using methods in [37].

3.2 Variational Characterizations of Hyperspherical Uniformity

While there are many feasible choices to measure hyperspherical uniformity, we seek to use variational characterizations due to its simplicity. As examples, we consider three simple characterizations: minimum hyperspherical energy [35] which is inspired by Thomson problem [52, 55] and measures the hyperspherical uniformity through Coulomb potential energy, maximum hyperspherical separation [38] which is inspired by Tammes problem [54] and captures the degree of hyperspherical uniformity by the smallest pairwise distance, and maximum gram determinant [38] which measures the hyperspherical uniformity with the volume of the formed parallelotope.

Minimum hyperspherical energy. MHE seeks to find an equilibrium state with minimum potential energy that distributes n electrons on a unit hypersphere as evenly as possible. Hyperspherical uniformity is characterized by minimizing the hyperspherical energy for n vectors $\boldsymbol{V}_n = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n \in \mathbb{R}^d\}$:

$$\min_{\{\hat{\boldsymbol{v}}_1, \cdots, \hat{\boldsymbol{v}}_n \in \mathbb{S}^{d-1}\}} \left\{ E_s(\hat{\boldsymbol{V}}_n) := \sum_{i=1}^n \sum_{j=1, j\neq i}^n K_s(\hat{\boldsymbol{v}}_i, \hat{\boldsymbol{v}}_j) \right\}, \quad K_s(\hat{\boldsymbol{v}}_i, \hat{\boldsymbol{v}}_j) = \begin{cases} \|\hat{\boldsymbol{v}}_i - \hat{\boldsymbol{v}}_j\|^{-s}, & s > 0 \\ -\|\hat{\boldsymbol{v}}_i - \hat{\boldsymbol{v}}_j\|^{-s}, & s < 0 \end{cases}, \tag{4}$$

where $\hat{\boldsymbol{v}}_i := \frac{\boldsymbol{v}_i}{\|\boldsymbol{v}_i\|}$ is the i-th vector projected onto the unit hypersphere. With $\mathcal{HU}(\hat{\boldsymbol{V}}) = -E_s(\hat{\boldsymbol{V}})$, we apply MHE to HUG and formulate the new objective as follows ($s_b = 2, s_w = -1$):

$$\min_{\{\hat{\boldsymbol{x}}_i\}_{i=1}^n, \{\hat{\boldsymbol{w}}_c\}_{c=1}^C} \mathcal{L}_{\text{MHE-HUG}} := \alpha \cdot E_{s_b}\big(\{\hat{\boldsymbol{w}}_c\}_{c=1}^C\big) - \beta \cdot \sum_{c=1}^C E_{s_w}\big(\{\hat{\boldsymbol{x}}_i\}_{i\in A_c}, \hat{\boldsymbol{w}}_c\big) \tag{5}$$

which can already be used as to train neural networks. The intra-class variability term in Eq. 5 can be relaxed to a lower bound such that we can instead minimize a simple upper bound of $\mathcal{L}_{\text{MHE-HUG}}$:

$$\mathcal{L}'_{\text{MHE-HUG}} = \alpha \cdot \sum_{c\neq c'} \|\hat{\boldsymbol{w}}_c - \hat{\boldsymbol{w}}_{c'}\|^{-2} + \beta' \cdot \sum_c \sum_{i\in A_c} \|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{w}}_c\| \geq \mathcal{L}_{\text{MHE-HUG}} \tag{6}$$

which is much more efficient to compute in practice and thus can serve as a relaxed HUG objective. Moreover, $\mathcal{L}_{\text{MHE-HUG}}$ and $\mathcal{L}'_{\text{MHE-HUG}}$ share the same minimizer. Detailed derivation is in Appendix F.

Maximum hyperspherical separation. MHS uses a maximum geodesic separation principle by maximizing the *separation distance* $\vartheta(\hat{\boldsymbol{V}}_n)$ (*i.e.*, the smallest pairwise distance in $\boldsymbol{V}_n = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n \in \mathbb{R}^d\}$): $\max_{\hat{\boldsymbol{V}}} \{\vartheta(\hat{\boldsymbol{V}}_n) := \min_{i \neq j} \|\hat{\boldsymbol{v}}_i - \hat{\boldsymbol{v}}_j\|\}$. Because $\vartheta(\hat{\boldsymbol{V}}_n)$ is another variational definition, we cannot naively set $\mathcal{HU}(\cdot) = \vartheta(\cdot)$. We define $\vartheta^{-1}(\hat{\boldsymbol{V}}_n) := \max_{i \neq j} \|\hat{\boldsymbol{v}}_i - \hat{\boldsymbol{v}}_j\|$ and HUG becomes

$$\max_{\{\hat{\boldsymbol{x}}_i\}_{i=1}^n, \{\hat{\boldsymbol{w}}_c\}_{c=1}^C} \mathcal{L}_{\text{MHS-HUG}} := \alpha \cdot \vartheta\big(\{\hat{\boldsymbol{w}}_c\}_{c=1}^C\big) - \beta \cdot \sum_{c=1}^C \vartheta^{-1}\big(\{\hat{\boldsymbol{x}}_i\}_{i \in A_c}, \hat{\boldsymbol{w}}_c\big), \tag{7}$$

which, by replacing intra-class variability with its surrogate, results in a more efficient form:

$$\mathcal{L}'_{\text{MHS-HUG}} := \alpha \cdot \min_{c \neq c'} \|\hat{\boldsymbol{w}}_c - \hat{\boldsymbol{w}}_{c'}\| - \beta \cdot \sum_c \max_{i \in A_c} \|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{w}}_c\| \tag{8}$$

which is a max-min optimization with a simple nearest neighbor problem inside. We note that $\mathcal{L}_{\text{MHS-HUG}}$ and $\mathcal{L}'_{\text{MHS-HUG}}$ share the same maximizer. Detailed derivation is in Appendix F.

Maximum gram determinant. MGD characterizes the uniformity by computing a proxy to the volume of the parallelotope spanned by the vectors. MGD is defined with kernel gram determinant:

$$\max_{\{\hat{\boldsymbol{v}}_1, \cdots, \hat{\boldsymbol{v}}_n \in \mathbb{S}^{d-1}\}} \log\det\big(\boldsymbol{G} := \big(K(\hat{\boldsymbol{v}}_i, \hat{\boldsymbol{v}}_j)\big)_{i,j=1}^n\big), \quad K(\hat{\boldsymbol{v}}_i, \hat{\boldsymbol{v}}_j) = \exp\big(-\epsilon^2 \|\hat{\boldsymbol{v}}_i - \hat{\boldsymbol{v}}_j\|^2\big) \tag{9}$$

where we use a simple Gaussian kernel with parameter ϵ and define $\boldsymbol{G}(\{\hat{\boldsymbol{v}}_1, \cdots, \hat{\boldsymbol{v}}_n\})$ as the kernel gram matrix for $\hat{\boldsymbol{V}}_n = \{\hat{\boldsymbol{v}}_1, \cdots, \hat{\boldsymbol{v}}_n\}$. With $\mathcal{HU}(\hat{\boldsymbol{V}}_n) = \det \boldsymbol{G}(\hat{\boldsymbol{V}}_n)$, we have the HUG objective as

$$\max_{\{\hat{\boldsymbol{x}}_i\}_{i=1}^n, \{\hat{\boldsymbol{w}}_c\}_{c=1}^C} \mathcal{L}_{\text{MGD-HUG}} := \alpha \cdot \log\det\big(\boldsymbol{G}(\{\hat{\boldsymbol{w}}_c\}_{c=1}^C)\big) - \beta \cdot \sum_c \det\big(\boldsymbol{G}(\{\hat{\boldsymbol{x}}_i\}_{i \in A_c}, \hat{\boldsymbol{w}}_c)\big) \tag{10}$$

where we drop $\log(\cdot)$ for the intra-class variability term to avoid numerical issues. With MGD, HUG has interesting geometric interpretation – it encourages the volume spanned by class proxies to be as large as possible and the volume spanned by intra-class features to be as small as possible.

3.3 Theoretical Insights and Discussions

There are many interesting theoretical questions concerning HUG, and this framework is highly related to a few topics in mathematics, such as tight frame theory [58], potential theory [30], sphere packing and covering [3, 13, 18]. The depth and breath of these topics are beyond imagination. In this section, we focus on discussing some highly related yet intuitive theoretical properties of HUG.

Theorem 4 (Order of Minimum Hyperspherical Energy) *If $d-1 > s > 0$ or $-2 > s > 0$ and $d \in \mathbb{N}$, we have that $\lim_{n\to\infty}\{n^{-2} \cdot \min_{\hat{\boldsymbol{V}}_n} E_s(\hat{\boldsymbol{V}}_n)\} = c(s,d)$ where $c(s,d)$ is a constant involving s, d.*

The result above shows that the leading term of the minimum energy grows of order $\mathcal{O}(n^2)$ as $n \to \infty$. Theorem 4 generally holds with a wide range of s for the Riesz kernel in hyperspherical energy. Moreover, the following result shows that MHS is in fact a limiting case of MHE as $s \to \infty$.

Proposition 2 (MHS is a Limiting Case of MHE) *Let $n \in \mathbb{N}, n \geq 2$ be fixed and $(\mathbb{S}^{d-1}, L_2)$ be a compact metric space. We have that $\lim_{s\to\infty}(\min_{\hat{\boldsymbol{V}}_n \subset \mathbb{S}^{d-1}} E_s(\hat{\boldsymbol{V}}_n))^{\frac{1}{s}} = (\max_{\hat{\boldsymbol{V}}_n \subset \mathbb{S}^{d-1}} \vartheta(\hat{\boldsymbol{V}}_n))^{-1}$.*

Proposition 3 *The HUG objectives in both Eq. 5 and Eq. 6 converge to simplex ETF when $2 \leq C \leq d+1$, converge to cross-polytope when $C = 2d$ and asymptotically converge to GNC as $C \to \infty$.*

Proposition 3 shows that HUG not only decouples GNC but also provably converges to GNC. Since GNC indicates that the CE loss eventually approaches to the maximizer of HUG, we now look into how the CE loss implicitly maximizes the HUG objective in a coupled way.

Proposition 4 *The CE loss is $\mathcal{L}_{CE} = \sum_{i=1}^n \log(1 + \sum_{j \neq c_i}^C \exp(\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle - \langle \boldsymbol{w}_{c_i}, \boldsymbol{x}_i \rangle))$ where n is the number of samples, $\boldsymbol{x}_i$ is the i-th sample with label c_i and $\boldsymbol{w}_c$ is the last-layer linear classifier for the j-th class. Bias is omitted for simplicity. $\mathcal{L}_{CE}$ is bounded by ($\rho = C-1$)*

$$\underbrace{\sum_{i=1}^n \sum_{j \neq c_i}^C \langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle}_{Q_1\text{: coupled IS and IV}} - \underbrace{\rho \sum_{i=1}^n \langle \boldsymbol{w}_{c_i}, \boldsymbol{x}_i \rangle}_{Q_2\text{: Inter-class Variability}} \leq \mathcal{L}_{CE} \leq \log\big(1 + \underbrace{\sum_{i=1}^n \sum_{j \neq c_i}^C \exp(\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle)}_{Q_3\text{: coupled IS and IV}} + \underbrace{\rho \sum_{i=1}^n \exp(\langle \boldsymbol{w}_{c_i}, \boldsymbol{x}_i \rangle)}_{Q_4\text{: Inter-class Variability}}\big).$$

From Proposition 4, we can see that the CE loss inherently couples intra-class variability (IV) and inter-class separability (IS). With normalized classifiers and features, we can see that Q_1 and Q_3 have the same minimum where $\boldsymbol{x}_i = \boldsymbol{w}_{c_i}$ and $\boldsymbol{w}_i, \forall i$ attain hyperspherical uniformity (Appendix I).

[39] has proved that the minimizer of a normalized form of the CE loss converges to hyperspherical uniformity. Built upon this nice result, we can easily extend it to the following theorem:

Theorem 5 (CE Asymptotically Converges to HUG's Maximizer) *Considering unconstrained features of C classes (each class has the same number of samples), with features and classifiers normalized on some hypersphere, we have that, for the minimizer of the CE loss, classifiers converge weakly to the uniform measure on $\mathbb{S}^{d-1}$ as $C \to \infty$ and features collapse to their corresponding classifiers. The minimizer of CE also asymptotically converges to the maximizer of HUG.*

Theorem 5 shows that the minimizer of the CE loss with unconstrained features [42] asymptotically converges to the maximizer of HUG (*i.e.*, GNC). Till now, we show that HUG shares the same optimum with CE (with hyperspherical normalization), while being more flexible for decoupling inter-class feature separability and intra-class feature variability. Therefore, we argue that HUG can be an excellent alternative for the widely used CE loss in classification problems.

The role of feature and class proxy norm. Both NC and GNC do not take the norm of feature and class proxy into consideration. HUG also assume both feature and class proxy norm are projected onto some hypersphere. Although dropping these norms usually improves generalizability [7, 8, 11, 34, 59], training neural networks with standard CE loss still yields different class proxy norms and feature norms. We hypothesize that this is due to the underlying difference among training data distribution of different classes. One empirical evidence to support this is that average feature norm of different classes is consistent across training under different random seeds (*e.g.*, average feature norm for digit 1 on MNIST stays the smallest in different run). [27, 36, 41] empirically show that feature norm corresponds to the quality of the sample, which can also viewed as a proxy to sample uncertainty. [45] theoretically shows that the norm of neuron weights (*e.g.*, classifier) matters for its Rademacher complexity. As a trivial solution to minimize the CE loss, increasing the classifier norm (if the feature is correctly classified) can easily decrease the CE loss to zero for this sample, which is mostly caused by the softmax function. Taking both feature and class proxy norm into account greatly complicates the analysis (*e.g.*, it results in weighted hyperspherical energy where the potentials between vectors are weighted) and seem to yield little benefit. We defer this to future investigation.

HUG as a general framework for designing loss functions. HUG can be viewed as an inherently decoupled way of designing new loss functions. As long as we design a measure of hyperspherical uniformity, then HUG enables us to effortlessly turn it into a loss function for neural networks. We show that, both theoretically and empirically, HUG delivers better loss functions than the CE loss.

4 Experiments and Results

Our experiments aims to demonstrate the empirical effectiveness of HUG, so we focus on the fair comparison to the popular CE loss under the same setting. Experimental details are in Appendix A.

4.1 Exploratory Experiments and Ablation Study

Different HUG variants. We compare different HUG variants and the CE loss on CIFAR-10 and CIFAR-100 with ResNet-18 [22]. Specifically, we use Eq. 6, Eq. 6 and Eq. 10 for MHE-HUG, MHS-HUG and MGD-HUG, respectively. The results are given in Table 1. We can observe that all HUG variants outperform the CE loss. Among all, MHE-HUG achieves the best testing accuracy with considerable improvement over the CE loss. We note that all HUG variants are used without the CE loss. The performance gain of HUG are actually quite significant, since the CE loss is currently a default choice for classification problems and serves as a very strong baseline.

Method	CIFAR-10	CIFAR-100
CE Loss	5.45	24.90
MHE-HUG	**5.03**	**23.50**
MHS-HUG	5.09	24.38
MGD-HUG	5.38	24.59

Table 1: Testing error (%) of HUG variants on CIFAR-10 and CIFAR-100.

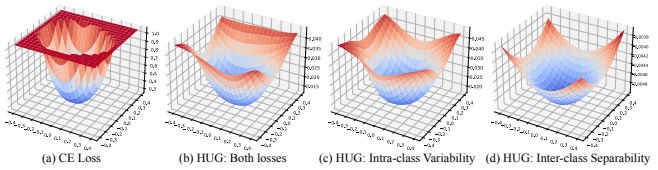
Different methods to update proxies. We also evaluate how different proxy update methods will affect the classification performance. We use the same setting as Table 1. For all the proxy update methods, we apply them to MHE-HUG (Eq. 6) under the same setting. The results are given Table 2. We can observe that all the propose proxy update methods work reasonably well. More interestingly, static proxies work surprisingly well and outperform the CE loss even when all the class proxies are randomly initialized and then fixed throughout the training. The reason the static proxies work for MHE-HUG

Method	CIFAR-10	CIFAR-100
CE Loss	5.45	24.90
Fully learnable	**5.03**	**23.50**
Static (random)	5.19	24.23
Static (optimized)	5.12	24.02
Partially learnable	5.08	23.89

Table 2: Testing error (%) of different proxy update methods on CIFAR-10 and CIFAR-100.

is due to Proposition 1. This result is significant since we no longer have to train class proxies in HUG (unlike CE). When trained with large number of classes, it is GPU-memory costly for learning class proxies, which is also known as one of the bottlenecks for face recognition [1]. HUG could be a promising solution to this problem.

Loss landscape and convergence. We perturb neuron weights (refer to [31]) to visualize the loss landscape of HUG and CE in Figure 4. We use MHE in HUG here. The results show that HUG yields much flatter local minima than the CE loss in general, implying that HUG has potentially stronger generalization [26, 46]. We show more visualizations and convergence dynamics in Appendix J.



(a) CE Loss (b) HUG: Both losses (c) HUG: Intra-class Variability (d) HUG: Inter-class Separability

Figure 4: Loss landscape visualization. (b,c,d) show $\mathcal{L}'_{\text{MHE-HUG}}$, T_b and T_w, respectively.

4.2 Generalization and Adversarial Robustness

Learning with different architectures. We evaluate HUG with different network architectures such as VGG-16 [51], ResNet-18 [22] and DenseNet-121 [23]. Results in Table 3 (Left number: CIFAR-10, right number: CIFAR-100) show that HUG is agnostic to different network architectures and outperforms the CE loss in every case. Although HUG works well on its own, any other methods that improve CE can also work with HUG.

Method	ResNet-18	VGG-16	DenseNet-121
CE Loss	5.45 / 24.90	5.28 / 22.99	5.04 / 21.47
HUG	**5.03 / 23.50**	**5.19 / 22.77**	**4.85 / 21.30**

Table 3: Testing error (%) with different architectures.

Long-tailed recognition. We consider the task of long-tailed recognition, where the data from different classes are imbalanced. The settings generally follow [5], and the dataset gets more imbalanced if the imbalance ratio (IR) gets smaller. The potential of HUG in imbalanced classification is evident, as the inter-class separability in the HUG is explicitly modeled and can be easily controlled. Experimental results in Table 4 show that HUG can consistently outperform the CE loss in the challenging long-tailed setting under different imbalanced ratio.

	CIFAR-100				CIFAR-10			
IR	0.2	0.1	0.02	0.01	0.2	0.1	0.02	0.01
CE	66.74	62.31	48.79	43.82	90.29	87.85	79.17	74.11
HUG	**67.83**	**63.33**	**50.48**	**45.63**	**90.41**	**88.20**	**79.88**	**75.14**

Table 4: Testing accuracy (%) of long-tailed recognition.

Continual learning. We also demonstrate the potential of HUG in the class-continual learning setting, where the training data is not sampled *i.i.d.* but comes in class by class. Since training data is highly biased, hyperspherical uniformity among class proxies is crucial. Due to the decoupled nature of HUG, we can explicitly increase the importance of inter-class separability, unlike the CE loss. We use a simple continual learning method – ER [48] where the CE loss with memory buffer is used. We simply replace it with HUG. Table 5 shows HUG is able to consistently improve the performance of ER method under different size of memory buffer.

	CIFAR-100			CIFAR-10		
Memory size	200	500	2000	200	500	2000
ER + CE	22.14	31.02	43.54	49.07	61.58	76.89
ER + HUG	**23.52**	**31.92**	**43.92**	**53.74**	**62.67**	**77.21**

Table 5: Final testing accuracy (%) of continual learning.

Adversarial robustness. We further test HUG's adversarial robustness. In our experiments, we consider the classical white-box PGD attack [40] on ResNet-18. The PGD attack iteration is set as 100 and the attack strength level is set as 2/255, 4/255, 8/255 in l_∞ norm. All networks are naturally training with either HUG or the CE loss. Results in Table 6 demonstrates that HUG yields consistently stronger adversarial robustness than the CE loss.

Method	Clean	l_∞=2/255	l_∞=4/255	l_∞=8/255
CE Loss	5.45 / 24.90	7.94 / 2.12	0.61 / 0	0 / 0
HUG	**5.03 / 23.50**	**15.24 / 5.26**	**3.45 / 1.24**	**1.76 / 0.44**

Table 6: Testing accuracy (%) under adversarial attacks.

5 Related Work and Concluding Remarks

We start by generalizing and decoupling the NC phenomenon, obtaining two basic principles for loss functions. Based on these principles, we identify a quantity hyperspherical uniformity gap, which not only decouples NC but also provides a general framework for designing loss functions. We demonstrate a few simple HUG variants that outperform the CE loss in terms of generalization and adversarial robustness. There is a large body of excellent work in NC that is related to HUG, such as [19, 25, 56, 64]. Different from existing work in hyperspherical uniformity [32, 35, 38] and generic diversity (decorrelation) [2, 6, 9, 43, 60, 62], HUG works as a new learning target (used without CE) rather than acting as a regularizer for the CE loss (used together with CE). Following the spirit of [24], we demonstrate the effectiveness of HUG as a valid substitute for CE.

REFERENCES

[1] Xiang An, Xuhan Zhu, Yuan Gao, Yang Xiao, Yongle Zhao, Ziyong Feng, Lan Wu, Bin Qin, Ming Zhang, Debing Zhang, et al. Partial fc: Training 10 million identities on a single machine. In *ICCV*, 2021.

[2] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? In *NeurIPS*, 2018.

[3] Sergiy V Borodachov, Douglas P Hardin, and Edward B Saff. *Discrete energy on rectifiable sets*. Springer, 2019.

[4] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

[5] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, volume 32, 2019.

[6] Tianlong Chen, Zhenyu Zhang, Yu Cheng, Ahmed Awadallah, and Zhangyang Wang. The principle of diversity: Training stronger vision transformers calls for reducing all levels of redundancy. In *CVPR*, 2022.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[8] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[9] Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. In *arXiv preprint arXiv:1511.06068*, 2015.

[10] Henry Cohn and Abhinav Kumar. Universally optimal distribution of points on spheres. *Journal of the American Mathematical Society*, 20(1):99–148, 2007.

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[13] D Jack Elzinga and Donald W Hearn. The minimum covering sphere problem. *Management science*, 19 (1):96–104, 1972.

[14] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2): 179–188, 1936.

[15] Eduardo García-Portugués and Thomas Verdebout. An overview of uniformity tests on the hypersphere. *arXiv preprint arXiv:1804.00286*, 2018.

[16] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[17] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, 2016.

[18] Thomas C Hales. The sphere packing problem. *Journal of Computational and Applied Mathematics*, 44 (1):41–76, 1992.

[19] XY Han, Vardan Papyan, and David L Donoho. Neural collapse under mse loss: Proximity to and dynamics on the central path. *arXiv preprint arXiv:2106.02073*, 2021.

[20] Doug P Hardin, Edward B Saff, et al. Discretizing manifolds via minimum energy points. *Notices of the AMS*, 51(10):1186–1194, 2004.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[24] Like Hui and Mikhail Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. In *ICLR*, 2021.

[25] Like Hui, Mikhail Belkin, and Preetum Nakkiran. Limitations of neural collapse for understanding generalization in deep learning. *arXiv preprint arXiv:2202.08384*, 2022.

[26] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[27] Minchul Kim, Anil K Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In *CVPR*, 2022.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[29] Arno Kuijlaars and E Saff. Asymptotics for minimal discrete energy on the sphere. *Transactions of the American Mathematical Society*, 350(2):523–538, 1998.

[30] NS Landkof. *Foundations of modern potential theory*, volume 180. Springer, 1972.

[31] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *NeurIPS*, 2018.

[32] Rongmei Lin, Weiyang Liu, Zhen Liu, Chen Feng, Zhiding Yu, James M Rehg, Li Xiong, and Le Song. Regularizing neural networks via minimizing hyperspherical energy. In *CVPR*, 2020.

[33] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.

[34] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017.

[35] Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. In *NeurIPS*, 2018.

[36] Weiyang Liu, Zhen Liu, Zhiding Yu, Bo Dai, Rongmei Lin, Yisen Wang, James M Rehg, and Le Song. Decoupled networks. In *CVPR*, 2018.

[37] Weiyang Liu, Rongmei Lin, Zhen Liu, James M Rehg, Liam Paull, Li Xiong, Le Song, and Adrian Weller. Orthogonal over-parameterized training. In *CVPR*, 2021.

[38] Weiyang Liu, Rongmei Lin, Zhen Liu, Li Xiong, Bernhard Schölkopf, and Adrian Weller. Learning with hyperspherical uniformity. In *AISTATS*, 2021.

[39] Jianfeng Lu and Stefan Steinerberger. Neural collapse under cross-entropy loss. *Applied and Computational Harmonic Analysis*, 59:224–241, 2022.

[40] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[41] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. Magface: A universal representation for face recognition and quality assessment. In *CVPR*, 2021.

[42] Dustin G Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *Sampling Theory, Signal Processing, and Data Analysis*, 20(2):1–13, 2022.

[43] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[44] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[45] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *COLT*, 2015.

[46] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *NIPS*, 2017.

[47] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[48] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

[49] Edward B Saff and Amo BJ Kuijlaars. Distributing many points on a sphere. *The mathematical intelligencer*, 19(1):5–11, 1997.

[50] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[52] Steve Smale. Mathematical problems for the next century. *The mathematical intelligencer*, 20(2):7–15, 1998.

[53] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *CVPR*, 2020.

[54] Pieter Merkus Lambertus Tammes. On the origin of number and arrangement of the places of exit on the surface of pollen-grains. *Recueil des travaux botaniques néerlandais*, 27(1):1–84, 1930.

[55] Joseph John Thomson. Xxiv. on the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 7(39):237–265, 1904.

[56] Christos Thrampoulidis, Ganesh R Kini, Vala Vakilian, and Tina Behnia. Imbalance trouble: Revisiting neural-collapse geometry. *arXiv preprint arXiv:2208.05512*, 2022.

[57] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[58] Shayne FD Waldron. *An introduction to finite tight frames*. Springer, 2018.

[59] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 2018.

[60] Jiayun Wang, Yubei Chen, Rudrasis Chakraborty, and Stella X Yu. Orthogonal convolutional neural networks. In *CVPR*, 2020.

[61] Yandong Wen, Weiyang Liu, Adrian Weller, Bhiksha Raj, and Rita Singh. Sphereface2: Binary classification is all you need for deep face recognition. In *ICLR*, 2022.

[62] Pengtao Xie, Aarti Singh, and Eric P Xing. Uncorrelation and evenness: a new diversity-promoting regularizer. In *ICML*, 2017.

[63] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[64] Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. In *NeurIPS*, 2021.