Hyperspherical Classification with Dynamic Label-to-Prototype Assignment

Mohammad Saeed Ebrahimi Saadabadi[1], Ali Dabouei[2], Sahar Rahimi Malakshan[1], Nasser M. Nasrabadi[3]

[1,3] West Virginia University, [2] Carnegie Mellon University

[1]{me00018, sr00033}@mix.wvu.edu, [2]adabouei@andrew.cmu.edu, [3]nasser.nasrabadi@mail.wvu.edu

Abstract

Aiming to enhance the utilization of metric space by the parametric softmax classifier, recent studies suggest replacing it with a non-parametric alternative. Although a non-parametric classifier may provide better metric space utilization, it introduces the challenge of capturing inter-class relationships. A shared characteristic among prior non-parametric classifiers is the static assignment of labels to prototypes during the training, i.e., each prototype consistently represents a class throughout the training course. Orthogonal to previous works, we present a simple yet effective method to optimize the category assigned to each prototype (label-to-prototype assignment) during the training. To this aim, we formalize the problem as a two-step optimization objective over network parameters and label-to-prototype assignment mapping. We solve this optimization using a sequential combination of gradient descent and Bipartite matching. We demonstrate the benefits of the proposed approach by conducting experiments on balanced and long-tail classification problems using different backbone network architectures. In particular, our method outperforms its competitors by 1.22% accuracy on CIFAR-100, and 2.15% on ImageNet-200 using a metric space dimension half of the size of its competitors. Code

1. Introduction

Image classification is a fundamental problem in deep learning [44]. Deep image classification models conventionally consist of a stack of non-linear feature extractors (backbone) together with a classification layer [18, 28, 53]. Despite diversified model design for their backbone, from convolutional neural networks [18, 26] to transformer architecture [11, 35], Parametric Softmax Classifier (PSC) with Cross-Entropy (CE) loss has been the *de facto* choice of design for the classification layer [18, 29, 51]. This widely adopted framework enables joint training of the PSC and backbone through gradient-based learning algorithms [4]. Yet, it suffers from several shortcomings [23, 42, 51, 54].
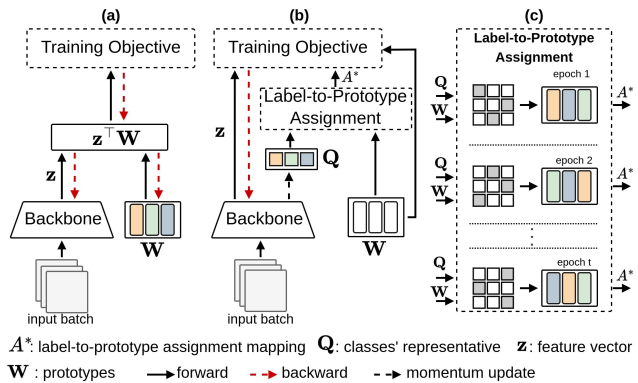

Figure 1. Comparison of the proposed method with the conventional PSC and the previous fixed classifier setup, using a toy example with three classes. Each color denotes a distinct class. a) Label-to-prototype assignment remains static during training. In PSC, optimization focuses on the network, consisting of the backbone and prototypes $\mathbf{W}$. In the case of a fixed classifier, only the backbone is optimized, and prototypes remain fixed. b) In the proposed method, prototypes within the hypersphere are fixed, and optimization targets the backbone and the label that each prototype represents. c) Toy example showing changes in label-to-prototype assignment during training.

First, it overlooks the intra-class compactness, focusing solely on the optimization of the relative intra and inter-class distances [42, 50]. Second, it fails to fully exploit the metric space, leading to a localized solution [12, 23, 33]. Third, the standard PSC enforces a fixed dimensionality of the output space, which hampers transferability and results in linear growth of the classifier parameter associated with the number of classes [47, 51]. Forth, when facing an unbalanced dataset *w.r.t.* sample per class, the PSC struggles with suboptimal performance for minority classes [10, 13, 54]. To address these issues, studies suggest employing auxiliary supervisions [12, 52], sample-based objectives [10, 46], distributed training [1, 2], active class selection [15], and loss reweighting [9, 24]. However, these methods concentrate exclusively on addressing a single limitation and do not generalize to all the issues [51].

Recently, studies have focused on replacing PSC with non-parametric solutions [39, 51, 54]. Yang *et al.* [54] employed a fixed Equiangular Tight Frame (ETF) geometrical structure as prototypes of the PSC providing maximal inter-class separation, optimal metric space exploitation, and a constant number of trainable parameters as the number of classes increases. Furthermore, fixed prototypes alleviate the problem of '*minority collapse*' in the learnable classifier trained on an imbalanced dataset [13]. However, a major issue with the ETF classifier [54] is that ETF prototypes force any pairs of classes to reflect equal similarity. Intuitively, semantically unrelated classes should be farther apart from the related ones, and equal inter-class similarity for all categories is counter-intuitive [25]. Besides, ETF requires the metric space dimensionality to be greater than the number of classes, limiting its applicability [36, 43].

In the quest for a more appropriate non-parametric classification alternative, Wang *et al.* [51] obtain fixed prototypes by summarizing each class into a cluster. Albeit effective for supervising intra-class compactness, it fails to fully exploit the metric space. Recently, Mettes *et al.* [39] utilized fixed equidistributed hyperspherical points as prototypes of PSC. While equidistributed prototypes enhance metric space exploitation [12, 23, 39], the question that naturally arises here is: *with fixed prototypes, how to define the label-to-prototype assignment to capture the dataset's inter-class relationships?* To answer this, HPN [39] utilizes the classes' names as privileged information; however, such privileged information can be inadequate or absent [14]. A common characteristic amongst previous learnable and fixed classifiers is that the label-to-prototype assignment is static, and each prototype represents a unique label throughout the training course, as depicted in Figure 1a.

In this study, our methodology diverges from static label-to-prototype assignment and focuses on optimizing the assignment throughout the training when the prototypes are fixed, as illustrated in Figure 1b. Therefore, the model changes the label-to-prototype assignment while the prototypes are pre-distributed in the hypersphere and fixed. Ideally, this ensures the maximal inter-class separation throughout fully exploiting the metric space in a manner consistent with the inter-class relationships, as depicted in Figure 1c. We divide the c-way classification into three subtasks: 1) organizing c equidistributed prototypes in the hypersphere, 2) solving a bipartite matching problem from the classes to the predefined prototypes, and 3) solving a regression problem from input to discrete approximation of unit hypersphere. The solution to the former depends merely on the metric space dimensionality and the number of classes. We formulate the second and final stages as a two-step optimization over the label-to-prototype assignment mapping and network parameters. The main contributions of this paper can be summarized as follows:

- We propose a novel classification framework in which the prototypes are fixed but the label-to-prototype assignment is adaptive and changes during the course of training.
- We formalize the problem as a two-step optimization and solve it using a combination of bipartite matching and gradient descent. This procedure does not use any auxiliary information to achieve this goal.
- We empirically show that our method is a generalized version of the ETF methods by removing the constraint on the number of classes.
- We evaluate our method on both balanced and long-tailed classification, achieving superior or comparable results compared to state-of-the-art fixed classifier methods while being more effective in training and scale.

2. Related Works

Deep Neural Networks (DNNs) have become widely used in machine learning, excelling in classification with end-to-end training using PSC [18, 28]. Motivated by the huge computational demand of the last linear layer [20, 47, 51], limited inter-class separation [12, 39, 42, 50], and poor classification accuracy of minority classes [10, 47, 51, 54], recent studies have focused on replacing the PSC with non-parametric alternatives [17, 20, 30, 38, 41]. To this end, a group of studies [36, 54, 55] focused on replacing prototypes of PSC with predefined fixed ETF. A set of c lines $\mathcal{L} = \{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_c\}$ through the origin of d-dimensional space is termed equiangular if there is a constant $\rho \geq 0$, such that $\langle \mathbf{l}_i, \mathbf{l}_j \rangle = \pm\rho$ for all $1 \leq i < j \leq c$ [56]. Here, $\langle ., . \rangle$ represents the inner product. While ETF provides optimal inter-class separation, we argue that it can be suboptimal from the perspective of inter-class relationships. Intuitively, unrelated classes should be farther apart from the related ones. However, an ETF classifier forces a separable but counter-intuitive structure in which all the classes have equal relationships. Moreover, while constructing simplex ETF for pair (d, c) is not limited to $d \geq (c - 1)$, studies demonstrate that ETF is only applicable to DNNs classification framework when $d \geq (c - 1)$, which critically limits its practical utilization [3, 48, 56].

To bypass these limitations, Wang *et al.* [51] adhere to the well-established idea of the nearest centroid classifier. The classifier in this framework is obtained by summarizing each class into a cluster. While effective in capturing inter-class relationships, the efficacy of metric space exploitation is overlooked [39]. Mettes *et al.* [39] proposed employing fixed equidistributed hyperspherical points as prototypes of PSC. Ideally, equidistributed prototypes provide full exploitation of the metric space [12, 39]. While promising, this method requires privileged information, to capture the inter-class relationships, which can be inadequate or absent [14]. Common among previous methods is that the label-to-prototype assignment is constant during training.

In an orthogonal direction, we propose to fix the prototype in an equidistributed organization and change the label-to-prototype assignment. Thus, the semantic category that every prototype represents is not unique during the training and changes *w.r.t.* the state of the feature extractor. For distributing the prototypes on the hypersphere, we optimized a cost function based on the Gaussian potential kernel, which is closely connected to universally optimal point configurations [7, 8], and we empirically show that our equidistributed answer converges to the ETF structure when the metric space dimensionality requirement is met.

3. Method

Notation. Let $\mathcal{D} = \{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{n}$ be the training set consisting of c classes and n samples. Suppose $M_\theta(.) : \mathcal{X} \rightarrow \mathbb{R}^d$ is the deep feature extractor that maps an input sample $\mathbf{x}_i$ to d-dimensional representation $\mathbf{z}_i \in \mathbb{R}^d$. $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_c] \in \mathbb{R}^{d \times c}$ is the matrix storing hyperspherical prototypes $\mathbf{w}_i \in \mathbb{R}^d$. $A : [1, 2, \ldots, c] \rightarrow [1, 2, \ldots, c]$ is the label-to-prototype assignment, denoting a bijective mapping from labels to prototypes. A is constant in conventional PSC and equiangular setup and optimizable in our method. $\mathbf{Q} = [\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2, \ldots, \bar{\mathbf{z}}_c] \in \mathbb{R}^{d \times c}$ is the matrix of the classes' representative, storing the momentum-averaged latent feature of each class during training. For the convenience of presentation, all the representations and prototypes are ℓ_2-normalized.

3.1. Overview

We are interested in learning a classification model that fully exploits the metric space while capturing the underlying characteristics of the target dataset. Inspired by the recent works on replacing the PSC with non-parametric alternatives [17, 20, 22, 30, 38, 41], we fix the equidistributed hyperspherical prototypes [47, 55] to maintain large-margin separation among class prototypes during the training. Hence, the supervised classification is generally defined as mapping input images to a discrete approximation of the unit hypersphere:

$$\min_\theta \sum_{i=1}^{n} L(M_\theta(\mathbf{x}_i), \mathbf{w}_{A(y_i)}), \tag{1}$$

where L is the training objective, tryings to align features $\mathbf{z}_i$ with their corresponding prototype $\mathbf{w}_{A(y_i)}$.

In the case of ETF [54], the assignment of prototypes is an arbitrary mapping that remains constant during training, as any given pair of prototypes exhibits equal similarity. While this assumption may hold for specific data, it fails to generalize, *e.g.*, for a multi-modal data distribution where different classes share similar properties. Thus, it is expected that the similarity between members of the same super-class to be higher than the similarity between
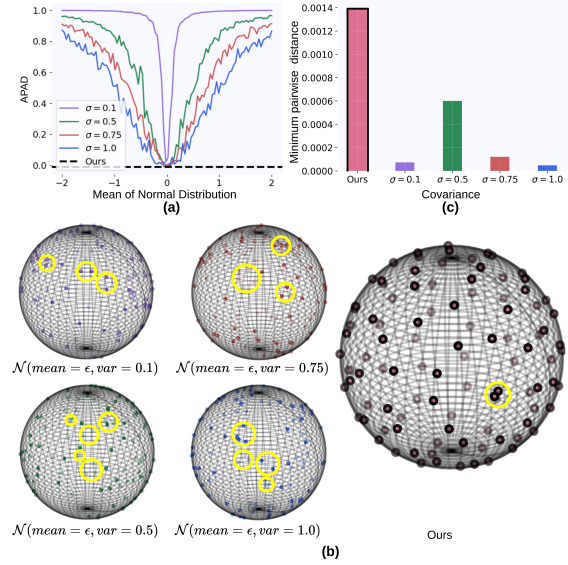

Figure 2. Comparing the Average Pairwise Angular Distance (APAD) value of the 100 prototypes drawn from multivariate Gaussian distributions with the covariance matrix of σI and the proposed optimized prototypes. a) Analysis of the APAD value. Notably, random prototypes drawn from arbitrary zero-mean distributions yield the optimal APAD value, underscoring that uniformity across S^{d-1} cannot be solely guaranteed by this objective function. b) Illustrating optimized and degenerate solutions on the S^2. Highlighted in yellow are areas where multiple prototypes exhibit closer proximity than ideal. c) Comparison of the minimum cosine distance, *i.e.*, $1 - \cos(\mathbf{w}_i, \mathbf{w}_j)$, for our optimized and degenerate solutions prototypes. Greater distances are indicative of superior utilization of metric space.

members of different super-classes [6], *e.g.*, as the '*truck*' and '*automobile*' classes in CIFAR-10. Also, in the case of PSC setup, A is the identity mapping $A(y_i) = y_i$ and fixed during the training, *i.e.*, each class is assigned to a specific trainable prototype. However, PSC ignores the distribution of features in holistic metric space and may lead to high locality [12, 39]. Here, A is not fixed during the training, and we aim to find the best prototype for each class from the predefined prototype set $\mathbf{W}$. Hence, our main objective is:

$$\min_\theta \min_A \sum_{i=1}^{n} L(M_\theta(\mathbf{x}_i), \mathbf{w}_{A(y_i)}). \tag{2}$$

Our overall framework involves an optimization problem *w.r.t.* two sets of variables θ, and A. This problem does not have a closed-form solution due to the highly non-linear nature of DNNs. Furthermore, A is not differentiable, and the whole framework cannot be optimized using gradient descent. We formulate a two-step optimization to solve this problem. In the first step, we solve a bipartite matching problem between classes' representatives $\mathbf{Q}$ and fixed prototypes $\mathbf{W}$, which is essential to capture the inter-class relationships in the metric space. In second step, we optimize L with respect to θ given A and $\mathbf{W}$.
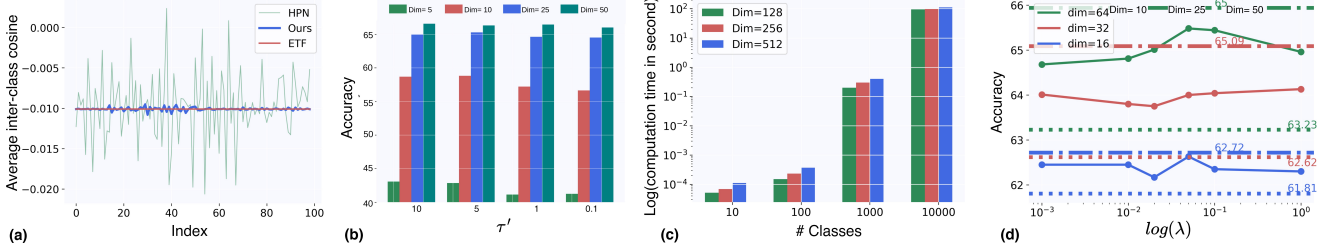
Figure 3. a) Average inter-prototype cosine when $d = c = 100$. b) Classification accuracy (%) on CIFAR-100 with ResNet-32 when the τ' changes. c) Time consumed for updating the A. Since we update the label-to-prototype assignment every epoch, $\tau' = 1.0$, this computation time is negligible compared to the total training time. d) Effect of regularizing the PSC with L_{uni} with different scaling hyperparameter λ. The horizontal dotted and '-.-' lines represent the PSC and proposed method performance, respectively.

Inspired by [49], we enhance the metric space utilization by discretizing the hypersphere with prototypes distributed as uniformly as possible prior to training. During the training, optimization in Equation 2 does not involve updating the prototypes and maintains the large-margin separation among classes. Albeit the predefined and fixed position of $\mathbf{W}$, our framework changes the corresponding label for each prototype to reduce the classification loss function. In this way, given well-separated prototypes, our approach assigns classes to prototypes to minimize the classification loss and capture inter-class dependencies rather than fixing class relationships similar to the ETF setup [54]. Furthermore, the label-to-prototype assignment is based on model learning with no need for privileged information [39]. Algorithm 1 provides an overview of the presented method.

3.2. Classification with Dynamic Label-to-Prototype Assignment

Here, we provide a detailed description of our method. Prior to training, we distribute hyperspherical prototypes as uniformly as possible in a dataset and architecture-independent manner. Given the optimized prototypes, we adopt a two-step algorithm where the goal is to find parameters θ and A that best describe the data distribution.

3.2.1 Hyperspherical Prototype Estimation

Metric space utilization of the PSC relies on its hyperspherical prototype distribution [12]. To enhance metric space utilization, studies tried to encourage the uniformity of prototypes' distribution in the hyperspherical space by adding regularization to conventional PSC setup [12, 32, 52]. Recently, Yang *et al*. [54] defined prototypes as vertices of a simplex ETF that provides maximal inter-class separability. However, the ETF structure of prototypes is a strong assumption in which all pairs of classes are forced to have equal similarity. Also, ETF requires $d \geq (c-1)$, impractical for large c [47, 54]. To maximize the metric space utilization, we distribute prototypes in the hypersphere as uniformly as possible in a data and architecture-independent manner and fix them during the network training.

For $d = 2$ and c prototypes, the problem of uniformly distributing prototypes on the unit hypersphere S^{d-1} reduces to splitting the unit circle into equal slices with $\frac{2\pi}{c}$ angles. However, no optimal solution exists for $d \geq 3$ [45]. As a viable alternative, we use a differentiable uniformity metric to encourage mapping prototypes to a uniform distribution over S^{d-1}. Designing an objective minimized by uniform distribution is not trivial [8]. For instance, Wang *et al*. [49] demonstrated that an arbitrary zero mean distribution is a degenerate solution of Average Pairwise Angular Distance (APAD) [7, 8], as depicted in Figure 2.

We built our uniformity objective upon recent progress in contrastive representation learning [49]. Let $p_{\mathbf{w}}(.)$ be the prototypes' distribution over S^{d-1}. $G_t(\mathbf{u}, \mathbf{v}) \triangleq S^{d-1} \times S^{d-1} \to \mathbb{R}_+$ is the Gaussian potential kernel:

$$G_t(\mathbf{u}, \mathbf{v}) \triangleq e^{-t||\mathbf{u}-\mathbf{v}||_2^2}; \quad t > 0, \tag{3}$$

the uniformity loss can be defined as:

$$L \triangleq \log \mathop{\mathbb{E}}_{\mathbf{w}_i, \mathbf{w}_j \overset{\text{i.i.d.}}{\sim} p_{\mathbf{w}}} [G_t(\mathbf{u}, \mathbf{v})]; \quad t > 0, \tag{4}$$

which is nicely tied with the uniform distribution of points on the S^{d-1} [5]; detailed derivation can be found in [49].

In practice, we first randomly initialize the prototype matrix $\mathbf{W} \in \mathbb{R}^{d \times c}$. At every iteration, we optimize a subset of $\mathbf{W}$ using the following loss function:

$$L_{uni} = \log \left(\frac{1}{\hat{c}} \sum_{i=1}^{\hat{c}} \sum_{j=1}^{c} g_{i,j} \right), \tag{5}$$

where $\widehat{\mathbf{W}} \in \mathbb{R}^{d \times \hat{c}}$ is a subset of $\mathbf{W}$ that randomly selects $\hat{c} < c$ prototypes, and $g_{i,j}$ is the element of $\mathbf{G}$ that reflects the pairwise Gaussian potential between prototypes $\mathbf{w}_i$ and $\mathbf{w}_j$. Subsequent to updating the prototypes, they are mapped to the unit hypersphere using ℓ_2-normalization.

Please note that the solution to the distribution of the prototypes is independent of the dataset and network architecture. Instead, it depends merely on the metric space dimensionality, d, and the number of classes, c. Once we obtain a solution for a specified d and c, this $\mathbf{W}$ can be used across

	CIFAR-100				ImageNet-200			
	$d = 10$	$d = 25$	$d = 50$	$d = 100$	$d = 25$	$d = 50$	$d = 100$	$d = 200$
PSC	25.67	60.0	60.6	62.1	-	-	-	33.1
Word2Vec [40]	29.0	44.5	54.3	57.6	20.7	27.6	29.8	30.0
HPN [39]	51.1	63.0	64.7	**65.0**	38.6	44.7	44.6	**44.7**
Ours	**57.21**	**64.63**	<u>66.22</u>	62.85	**41.71**	**46.57**	<u>46.85</u>	37.28

Table 1. Classification accuracy (%) of our method and baseline prototypical methods using ResNet-32. Please note that our best result across dimensions (underlined), is always better than the best performance of baseline methods. Our method obtains the best accuracy across dimensions and datasets when $d < c$, emphasizing the role of dynamic label-to-prototype mapping in metric space exploitation.

Dataset	PSC	Word2Vec [40]	HPN [39]	ETF [54]	Ours
CIFAR-100	62.1	57.6	65.0	65.34	**66.22**
ImageNet-200	33.1	30.0	44.7	37.15	**46.85**

Table 2. Comparison of classification accuracy (%) of the proposed method with $d = \frac{c}{2}$ and other prototypical methods with $d = c$ using ResNet-32.

problems with the same number of categories and the same metric space dimensionality. For instance, in Table 6 the same $\mathbf{W}$ is used for CIFAR-10, SVHN, and STL-10.

3.2.2 Input-to-Prototype Mapping

As pointed out by [10, 54], in the training of PSC, the '*pull-push*' mechanism of Cross-Entropy (CE) implicitly forces intra-class compactness and inter-class separability, please see Section 6 of Supplementary Material for detailed analysis on CE derivative. Our method fixes the prototypes as the solution for inter-class separability, and the model does not learn it during the training. Hence, for a training sample $\mathbf{x}_i$ we seek to learn a mapping from the input to its assigned prototype, *i.e.*, '*pull*' $\mathbf{z}_i$ toward $\mathbf{w}_{A(y_i)}$:

$$L_{IPM}(\mathbf{z}_i, \mathbf{w}_{A(y_i)}) = \frac{1}{2}(\mathbf{z}_i^\top \mathbf{w}_{A(y_i)} - 1)^2, \quad (6)$$

this loss is only concerned with solving a regression task between the $\mathbf{z}_i$ and $\mathbf{w}_{A(y_i)}$; supervising the intra-class compactness by decreasing the angular distance of samples and their assigned prototypes.

Unlike the CE loss, Equation 6 is derived using a single prototype per sample. This approach reduces the computational workload compared to CE, where all prototypes are required for every sample [1]. Since we do not update the prototype during the training, our approach requires a backpropagation step through the M_θ. The partial derivative of Equation 6 can be given as:

$$\frac{\partial L_{IPM}}{\partial \mathbf{z}_i} = -(1 - \mathbf{z}_i^\top \mathbf{w}_{A(y_i)})\mathbf{z}_i^\top \mathbf{w}_{A(y_i)}, \quad (7)$$

which is identical to the '*pull*' force in the CE derivative. Since the Equation 7 is only concerned with aligning the samples to their assigned prototype, the problem of frequent *passive update* [10] of minority classes is relieved for

long-tail datasets [13]. Ideally, the combination of Equation 6 and fixed equidistributed prototypes, increases the intra-class compactness in an optimally separate prototype organization, *i.e.*, maximal inter-class separation.

3.2.3 Bijective Label-to-Prototype Mapping

Our proposal hinges on finding suitable label-to-prototype assignment A during the training. In the ETF setup, A is initialized randomly. In the case of PSC setup, the assignment is identity mapping. Note that in both these scenarios, A remains constant during the training. Here, our objective is to find assignment mapping that maximizes the log-likelihood function of the observed class:

$$A^* = \arg\min_A \sum_{j=1}^{c} -\log \frac{e^{\bar{\mathbf{z}}_j^\top \mathbf{w}_{A(j)}}}{\sum_{k=1}^{c} e^{\bar{\mathbf{z}}_j^\top \mathbf{w}_{A(k)}}}$$
$$= \arg\min_A \sum_{j=1}^{c} \left[-\bar{\mathbf{z}}_j^\top \mathbf{w}_{A(j)} + \log \sum_{k=1}^{c} e^{\bar{\mathbf{z}}_j^\top \mathbf{w}_{A(k)}} \right], \quad (8)$$

where $\bar{\mathbf{z}}_{y_i} = \alpha\bar{\mathbf{z}}_{y_i} + (1 - \alpha)\mathbf{z}_i$ is the momentum averaging over the representations of each class and $\alpha \in [0, 1]$ is the momentum coefficient [19, 37]. The term $\log \sum_{k=1}^{c} e^{\bar{\mathbf{z}}_j^\top \mathbf{w}_{A(k)}}$ is constant for all elements of A. Hence, the optimization term becomes:

$$A^* = \arg\min_A \sum_{j=1}^{c} -\bar{\mathbf{z}}_j^\top \mathbf{w}_{A(j)}, \quad (9)$$

following [21, 31] we solve this bipartite matching problem using Hungarian algorithm [27]. The resulting A^* is used to optimize Equation 6. We update the prototypes' assignment every τ iterations to sync it with model learning.

4. Experiment

As discussed in Section 3.2.2, employing fixed prototypes and L_{IPM} is particularly effective for the long-tail classification problem, also shown in [10, 13]. Hence, we conduct experiments in balanced and long-tailed setups to illustrate the efficacy of the proposed method in both scenarios.

Algorithm 1: Classification with dynamic Label-to-prototype assignment

---

1 Initialize A, $\mathbf{W}$, θ, $\mathbf{Q}$, $t_1 > 0$, and $t_2 > 0$.
2 **for** $t = 0 \dots t_1 - 1$ **do**
3   Compute L_{uni}, $\forall \mathbf{w}_i \in \mathbf{W}$
4   Update $\mathbf{W}$ using $\nabla_{\mathbf{W}} L_{uni}$ $\triangleright$ prototype estimation
5 **end**
6 **for** $t = 0 \dots t_2 - 1$ **do**
7   **for** *Batch in Dataset* **do**
8 $\mathbf{z} = M(Batch)$
9   Update $\mathbf{Q}$ using momentum averaging the representations of each class
10   Compute L_{IPM} using $\mathbf{z}$, A, and $\mathbf{W}$
11   Update M_θ using $\nabla_\theta L_{IPM}$; $\triangleright$ backbone training
12   **end**
13   Update assignment A using Equation 9;
14 **end**

4.1. Implementation Details

For precomputing the prototypes, we used SGD optimizer with a constant learning rate of 0.1, and the optimization was performed for 1000 iterations with the batch size equal to the number of classes. We conduct evaluations on balanced ImageNet-1K, ImageNet-200, and CIFAR-100 using ResNet-32, ResNet-50 [18], and Swin-Transformer [35] architectures. We train ResNet-32 and ResNet-50 using the SGD optimizer, and Swin-T is trained from scratch using Adam-W. For the sake of a fair comparison in balanced evaluations, we followed [39] for data augmentation. We conduct long-tailed evaluations on CIFAR-10, CIFAR-100, SVHN, and STL-10 using ResNet-32 and on ImageNet-LT [34] using ResNet-50. The models are trained from scratch using the SGD optimizer. The detailed experimental setup is described in Section 7 of Supplementary Material.

4.2. Balanced Classification

We compare our method with three different baselines of PSC, Word2Vec [40] and HPN [39]. Table 1 presents results for evaluations on ImageNet-200 and CIFAR-100. For both datasets, our method obtains the highest accuracy across metric space dimensions and datasets. This shows the better metric space exploitation by our predefined prototypes which are distributed based on Gaussian potential. Furthermore, these results demonstrate the superiority of our method in optimizing label-to-prototype assignment based on model training. Note that HPN utilizes privileged information of word embedding of classes' names which is not related to the training and comes from an independent model. However, we optimize the assignment based on model learning without employing any auxiliary information. We also observe that the superiority of the proposed method is more remarkable when the metric space dimension is significantly lower than the number of classes $d \ll c$. It can be attributed to the dynamic label-to-prototype as-

Method	Venue	Backbone	Optimizer	Accuracy (%)
PSC [18]	CVPR2016	ResNet-50	SGD	76.51
DNC [51]	ICLR2022	ResNet-50	SGD	76.49
Goto *et al.* [16]	WACV2024	ResNet-50	SGD	77.19
Kasarla *et al.* [23]	NEURIPS2022	ResNet-50	SGD	74.80
Ours	CVPR2024	ResNet-50	SGD	**77.47**
DNC [51]	ICLR2022	ResNet-101	SGD	77.80
Goto *et al.* [16]	WACV2024	ResNet-101	SGD	78.27
Kasarla *et al.* [23]	NEURIPS2022	ResNet-152	SGD	78.5
Ours	CVPR2024	ResNet-101	SGD	**79.63**
PSC	CVPR2016	Swin-T	AdamW	76.91
Ours	CVPR2024	Swin-T	AdamW	**77.26**

Table 3. Classification accuracy (%) on ImageNet-1K. The training was performed for 100 epochs when $d = 512$.

signment that captures inter-class relationships which are more noticeable when $d \ll c$.

Moreover, we observe that our method achieves the best performance across metric space dimensions. Specifically, in CIFAR-100, our approach attains an accuracy of 66.22% in $\mathbb{R}^{50}$, surpassing HPN's best performance by 1.22%, regardless of dimension. Similarly, in ImageNet-200, our method in $\mathbb{R}^{100}$ outperforms HPN's best performance across metric space dimensions by 2.14%. This emphasizes the role of label-to-prototype assignment which is an optimization variable in our method and constant in other fixed classifier baselines. When $d = c$ our optimized prototypes converge toward ETF which leads to the counter-intuitive solution in that all classes have equal similarity, causing our best performance to be achieved when $d < c$.

In terms of robustness to changes in metric space dimension, HPN experiences a performance drop of 11.89% and 1.7% on CIFAR-100 when transitioning from $25 \to 10$ and $50 \to 25$, respectively. In comparison, our method demonstrates greater stability with performance degradations of 7.41% and 1.6%. Similarly, in ImageNet-200, when transitioning from $50 \to 25$, HPN experiences a 6.1% performance drop, while our method is more robust with a performance degradation of only 4.85%. This performance robustness demonstrates that despite the reduction in metric space dimension and inter-class separability, our label-to-prototype assignment successfully captures inter-class relationships, leading to the capability to handle classification with much fewer metric space dimensions, *i.e.*, better metric space exploitation.

Table 2 compares the results of equidistributed prototypes, *i.e.*, Word2Vec [40], HPN [39], and our approach, with ETF [54] on CIFAR-100 and ImageNet-200 evaluations. In these evaluations, other methods use $d = c$ (the best performance of baseline methods), while our method employs $d = \frac{c}{2}$. Please note that the ETF cannot implemented in $\mathbb{R}^{\frac{c}{2}}$ since the constraint for ETF requires $d \geq (c - 1)$. We observe that the presented method outperforms ETF structure by 0.87% and 9.15% in CIFAR-100 and ImageNet-200, respectively. The results highlight the effectiveness of our approach in lifting the dimensional-

Method	d	0.005	0.01	0.02
PSC	128	38.7	43.0	48.1
ETF	128	<u>40.9</u>	**45.3**	50.4
Ours	128	**41.3**	44.9	<u>50.7</u>
Ours	50	<u>40.9</u>	<u>45.2</u>	**50.8**

Table 4. Long-tailed CIFAR-100 classification accuracy (%) of the proposed and baseline prototypical methods using ResNet-32. Note that our method using $d = 50$ could get the best or second-best performance in all the experiments.

PSC	ETF ($d = 4096$)	Ours ($d = 4096$)	Ours ($d = 512$)
44.3	44.7	44.6	**44.9**

Table 5. ImageNet-LT classification accuracy (%) of the proposed and baseline prototypical methods using ResNet-50 as backbone.

	CIFAR-10			SVHN			STL-10		
	0.005	0.01	0.02	0.005	0.01	0.02	0.005	0.01	0.02
PSC	67.3	72.8	78.6	40.5	40.9	49.3	33.1	**37.9**	**38.8**
ETF	**71.9**	76.5	81.0	**42.8**	45.7	**49.8**	33.5	37.2	37.9
Ours	71.5	**76.9**	**81.4**	40.9	**47.0**	49.7	**35.7**	35.6	38.0

Table 6. Long-tailed classification accuracy (%) of the proposed and baseline prototypical methods using ResNet-32 when $d = 64 > c$ for imbalance factors of 0.02, 0.01, and 0.005. Note that the same $\mathbf{W}$ is used for these evaluations. On-par results with ETF illustrate the convergence of our optimized prototype toward ETF structure when $d > c$.

	$d = 100$		$d = 50$	
	Max	Min	Max	Min
Word2Vec	0.74	-0.32	-	-
HPN	0.05	**-0.39**	-0.05	-0.62
ETF	**0.0**	-0.01	-	-
Ours	**0.0**	-0.01	**0.01**	**-1.0**

Table 7. The maximum ↓, and minimum ↓ pairwise cosine similarity among 100 prototypes. The proposed distribution of prototypes obtains the best scores when $d < c$ and converges toward ETF geometrical structure when $d = c$.

ity constraint of the ETF and outperforming baselines with less metric space dimension. Our results on ImageNet-1K are presented in Table 3, showing the superior performance of the presented method when compared to the conventional PSC framework. These improvements signify that our approach is not restricted to small-scale classification like CIFAR-100 or architecture like ResNet-32.

4.3. Long-tail Classification

The standard PSC framework with Cross-Entropy (CE) loss suffers from frequent '*passive updates*' of minority classes, leading to low separability among them [10]. Table 4 presents the results for long-tail evaluations on CIFAR-100. Our method consistently outperforms the standard PSC classifier across various imbalance factors, demonstrating its ability to alleviate the issue of frequent '*passive updates*' of minority classes. Furthermore, our approach advances the performance of the ETF without any constraint on the metric space dimension, underscoring its capability to overcome the ETF constraint on the number of classes and metric space dimensionality.

In Table 5, the advantage of the proposed method is further verified using ResNet-50 on ImageNet-LT. Concretely, our method outperforms PSC and ETF classifiers without having limitations on the metric space dimensionality and number of classes. These observations reflect the important role of dynamic label-to-prototype assignment in increasing metric space exploitation. Also, these observations show that our method is not limited to small-scale classification or backbone. Table 6 presents the results for evaluations on CIFAR-10, SVHN, and STL-10. Since the d and c are constant across these experiments, the same $\mathbf{W}$ is used for these evaluations. The on-par performance of our method with the ETF classifier indicates that our approach to distributing prototypes on the hypersphere converges toward the ETF organization of prototypes, *i.e.*, maximal separation, when the requirement of ETF is provided, *i.e.*, $d > c$.

4.4. Ablations

4.4.1 Hyperspherical Prototype Distribution

Prior to training, we distribute prototypes on S^{d-1}, and they remain fixed during the training. Hence, their separability critically affects the final discriminative power of the network. Following [39], we assess the distribution of prototypes by analyzing the maximum, and minimum pairwise cosine similarity among prototypes in Table 7. This analysis is conducted with 100 prototypes when $d = 50$ and $d = 100$. In $\mathbb{R}^{50}$, our optimized prototypes achieve the best scores, reflecting the effectiveness of our approach in evenly distributing points on the hypersphere and providing large-margin separation among prototypes. Note that the dimensionality constraint of ETF requires that the metric space dimension should be at least equal to the number of prototypes. Therefore, the row for 100 ETF prototypes in $\mathbb{R}^{50}$ is empty. In $\mathbb{R}^{100}$ our prototypes achieve the same scores as ETF, denoting the convergence of our method toward the ETF structure when $d \geq c$.

In Figure 3a, the advantage of our method in distributing hyperspherical prototypes is further verified by illustrating the average pairwise cosine of 100 prototypes when $d = c$. The distribution of prototypes in our method obtains a near-uniform distribution of inter-class cosine, similar to ETF, while HPN [39] demonstrates a high level of variation which illustrates the efficacy of our method in uniformly distributing prototypes. It is worth noting that the minimal cosine similarity of the c equiangular vectors in $\mathbb{R}^d$ is $\frac{-1}{c-1}$ [43], *i.e.*, $\frac{-1}{100-1} \approx -0.01$ in this experiment.
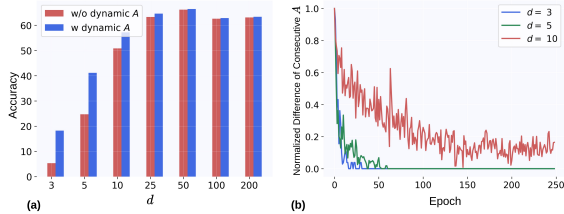
Figure 4. a) Classification accuracy (%) on CIFAR-100 using ResNet-32 w/o dynamic assignment showing the significance of optimizing A in low dimensional metric space. b) Normalized difference of consecutive assignments during training.

4.4.2 Ablation on τ

Here, we study the effect of the τ on the model performance. For the sake of brevity, we define $\tau' = \frac{\tau}{\lfloor \frac{n}{\text{batch size}} \rfloor}$ which denotes the number of assignment updates per each epoch. Figure 3b shows the classification accuracy on CIFAR-100 when τ' changes from 0.1 to 10. We observe that $\tau' = 10$ leads to performance improvement. However, with the increase in the d, the performance gap between $\tau' = 10$ and $\tau' = 1.0$ becomes reasonably close. Hence, to balance the computational efficacy and performance in the main experiments, we adhere to $\tau' = 1$.

4.4.3 Computational Time

Computing label-to-prototype assignment imposes an additional workload that grows non-linearly with the increase in the number of classes, *i.e.*, $O(c^3)$. Figure 3c reports the time consumed by a single label-to-prototype assignment computation. The reported value is the average of four runs. In our main experiments, we update the assignment mapping, A, at the end of every epoch, *i.e.*, $\tau = \lfloor \frac{n}{\text{batch size}} \rfloor$. Hence, acquiring label-to-prototype assignment incurs negligible runtime overhead at the scale of 10,000 classes.

4.4.4 Impact of Prototype Distribution

Previous studies [12, 32, 52] have employed the prototype uniformity constraint as the regularization to the Cross-Entropy (CE). In Table 8, we compare our method with the regularized version of PSC, *i.e.*, PSC*: CE + $\lambda \, L_{uni}$. As expected, PSC* improves the PSC baseline performance. However, it introduces the challenge of finding optimal tuning parameter λ, which can vary across metric space dimensionality, depicted in Figure 3d. Additionally, we employed the best-performing PSC prototypes as the fixed $\mathbf{W}$ of our method (referred to as Ours*). We observe that this model outperforms the baseline PSC model which suggests that our dynamic prototype matching provides notable enhancement to the training setup. Also, the intra-class compactness supervision of L_{IPM} is important for this improvement. Note that our main model outperforms all these baselines due to the better exploitation of metric space.

	$d=16$	$d=32$	$d=64$
PSC	62.62	61.81	63.23
PSC*	62.54	64.18	64.53
Ours*	60.51	63.43	64.71
Ours	**62.72**	**65.09**	**65.94**

Table 8. Abblation on the prototype distribution. The classification accuracy (%) is reported on CIFAR-100 using ResNet-32. PSC*: CE$+10^{-2} \times L_{uni}$. Ours*: substituting the equidistributed prototypes with softmax prototypes from a pre-trained network.

4.4.5 Impact of A

Figure 4a demonstrates the effect of label-to-prototype assignment in CIFAR-100 evaluation when d varies from 3 to 200. When $d \ll c$, we observe that our optimizable assignment A provides a notable enhancement to the training setup which emphasizes the importance of maintaining inter-class dependencies. As d increases, the effect of inter-class relationships fades. Also, we observe this effect in Figure 4b. With the increase in d, the label-to-prototype mapping is converging to a constant mapping. These observations reflect that our method lifts the dimensionality constraint of ETF and converges toward ETF when the metric space requirements are provided. Thus, as the prototype distribution gradually converges toward ETF organization, the similarity between every pair of prototypes becomes equal, and inter-class dependencies become less important.

5. Conclusion

In this paper, we proposed a dynamic label-to-prototype mapping to address the limitations of the current non-parametric alternative to the conventional Softmax classifier. Our method distributes prototypes before training and fixes them during training, to maintain the large-margin separation among prototypes and better metric space exploitation. Instead, we define the label-to-prototype assignment as an optimization variable, allowing the model to change the class that each prototype represents during training. The efficacy of the proposed method is demonstrated through various experiments on balanced/long-tail classification tasks and ablation studies.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1445–1449, 2021. 1, 5

[2] Xiang An, Jiankang Deng, Jia Guo, Ziyong Feng, XuHan Zhu, Jing Yang, and Tongliang Liu. Killing two birds with one stone: Efficient and robust training of face recognition cnns by partial fc. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4042–4051, 2022. 1

[3] Igor Balla, Felix Dräxler, Peter Keevash, and Benny Sudakov. Equiangular lines and spherical codes in Euclidean space. *Inventiones mathematicae*, 211(1):179–212, 2018. 2

[4] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision*, pages 456–473, 2018. 1

[5] Salomon Bochner. Monotone funktionen, stieltjessche integrale und harmonische analyse. *Mathematische Annalen*, 108(1):378–410, 1933. 4

[6] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *International Conference on Machine Learning*, pages 517–526. PMLR, 2017. 3

[7] Sergiy V Borodachov, Douglas P Hardin, and Edward B Saff. *Discrete energy on rectifiable sets*. Springer, 2019. 3, 4

[8] Henry Cohn and Abhinav Kumar. Universally optimal distribution of points on spheres. *Journal of the American Mathematical Society*, 20(1):99–148, 2007. 3, 4

[9] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 1

[10] Jiankang Deng, Jia Guo, Jing Yang, Alexandros Lattas, and Stefanos Zafeiriou. Variational prototype learning for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11906–11915, 2021. 1, 2, 5, 7

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[12] Yueqi Duan, Jiwen Lu, and Jie Zhou. Uniformface: Learning deep equidistributed representation for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3415–3424, 2019. 1, 2, 3, 4, 8

[13] Cong Fang, Hangfeng He, Qi Long, and Weijie J Su. Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. *Proceedings of the National Academy of Sciences*, 118(43):e2103091118, 2021. 1, 2, 5

[14] Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic busemann learning with ideal prototypes. *Advances in Neural Information Processing Systems*, 34:103–115, 2021. 2

[15] Joshua Goodman. Classes for fast maximum entropy training. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, pages 561–564. IEEE, 2001. 1

[16] Jumpei Goto, Yohei Nakata, Kiyofumi Abe, Yasunori Ishii, and Takayoshi Yamashita. Learning intra-class multimodal distributions with orthonormal matrices. In *WACV*, pages 1870–1879, 2024. 6

[17] Samantha Guerriero, Barbara Caputo, and Thomas Mensink. Deepncm: Deep nearest class mean classifiers. 2018. 2, 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2, 6

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 5

[20] Elad Hoffer, Itay Hubara, and Daniel Soudry. Fix your classifier: the marginal value of training the last weight layer. *arXiv preprint arXiv:1801.04540*, 2018. 2, 3

[21] Xiaowei Hu, Xi Yin, Kevin Lin, Lei Zhang, Jianfeng Gao, Lijuan Wang, and Zicheng Liu. Vivo: Visual vocabulary pre-training for novel object captioning. In *proceedings of the AAAI conference on artificial intelligence*, pages 1575–1583, 2021. 5

[22] Saumya Jetley, Bernardino Romera-Paredes, Sadeep Jayasumana, and Philip Torr. Prototypical priors: From improving classification to zero-shot learning. *arXiv preprint arXiv:1512.01192*, 2015. 3

[23] Tejaswi Kasarla, Gertjan Burghouts, Max van Spengler, Elise van der Pol, Rita Cucchiara, and Pascal Mettes. Maximum class separation as inductive bias in one matrix. *Advances in Neural Information Processing Systems*, 35: 19553–19566, 2022. 1, 2, 6

[24] Salman Khan, Munawar Hayat, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. Striking the right balance with uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 103–112, 2019. 1

[25] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 2

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[27] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 5

[28] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014. 1, 2

[29] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 1

[30] Xiao Li, Min Fang, Dazheng Feng, Haikun Li, and Jinqiao Wu. Prototype adjustment for zero shot classification. *Signal Processing: Image Communication*, 74:242–252, 2019. 2, 3

[31] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3694–3702, 2021. 5

[32] Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. *Advances in neural information processing systems*, 31, 2018. 4, 8

[33] Weiyang Liu, Zhen Liu, Zhiding Yu, Bo Dai, Rongmei Lin, Yisen Wang, James M Rehg, and Le Song. Decoupled networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2771–2779, 2018. 1

[34] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2537–2546, 2019. 6

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1, 6

[36] Jianfeng Lu and Stefan Steinerberger. Neural collapse with cross-entropy loss. *arXiv preprint arXiv:2012.08465*, 2020. 2

[37] James M Lucas and Michael S Saccucci. Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics*, 32(1):1–12, 1990. 5

[38] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013. 2, 3

[39] Pascal Mettes, Elise Van der Pol, and Cees Snoek. Hyperspherical prototype networks. *Advances in neural information processing systems*, 32, 2019. 2, 3, 4, 5, 6, 7

[40] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013. 5, 6

[41] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE international conference on computer vision*, pages 360–368, 2017. 2, 3

[42] Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, and Jun Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. *arXiv preprint arXiv:1905.10626*, 2019. 1, 2

[43] 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. 2, 7

[44] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017. 1

[45] Edward B Saff and Amo BJ Kuijlaars. Distributing many points on a sphere. *The mathematical intelligencer*, 19:5–11, 1997. 4

[46] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 1

[47] Yang Shen, Xuhao Sun, and Xiu-Shen Wei. Equiangular basis vectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11755–11765, 2023. 1, 2, 3, 4

[48] Mátyás A Sustik, Joel A Tropp, Inderjit S Dhillon, and Robert W Heath Jr. On the existence of equiangular tight frames. *Linear Algebra and its applications*, 426(2-3):619–635, 2007. 2

[49] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020. 4

[50] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7303–7313, 2021. 1, 2

[51] Wenguan Wang, Cheng Han, Tianfei Zhou, and Dongfang Liu. Visual recognition with deep nearest centroids. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2, 6

[52] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 499–515. Springer, 2016. 1, 4, 8

[53] Yibo Yang, Zhisheng Zhong, Tiancheng Shen, and Zhouchen Lin. Convolutional neural networks with alternately updated clique. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2413–2422, 2018. 1

[54] Yibo Yang, Shixiang Chen, Xiangtai Li, Liang Xie, Zhouchen Lin, and Dacheng Tao. Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network? *Advances in Neural Information Processing Systems*, 35:37991–38002, 2022. 1, 2, 3, 4, 5, 6

[55] Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class incremental learning. *arXiv preprint arXiv:2302.03004*, 2023. 2, 3

[56] Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34:29820–29834, 2021. 2