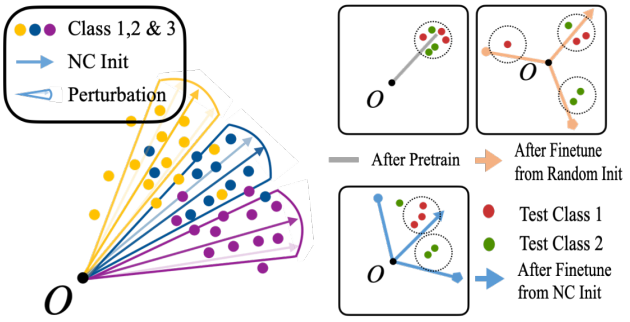
Neural Collapse-Informed Initialization with Perturbation Injection in Classification-based Metric Learning

Jinhee Park[13*], Hee Bin Yoo[2*], Minjun Kim[1], Byoung-Tak Zhang[2], Junseok Kwon[1†],
[1]Chung-Ang University, Korea
[2]Seoul National University, Korea
[3]Korea Electronics Technology Institute, Korea
iv4084em@cau.ac.kr, yooheebin@snu.ac, ekdh635@cau.ac.kr, btzhang@snu.ac.kr, jskwon@cau.ac.kr

Abstract

Recent studies have revealed Neural Collapse (NC) in deep classifiers, where last-layer weights and features align into an equiangular tight frame (ETF), concentrating class information along specific embedding directions. However, conventional fine-tuning typically disregards this structure, initializing task-specific classifier heads randomly. To explicitly leverage this phenomenon, we propose a simple yet effective method for metric learning: (1) initializing the classifier head along each class's NC direction from a pretrained model to preserve the emergent structure, and (2) injecting small isotropic Gaussian noise during finetuning to boost generalization. In addition, we provide a theoretical bound proving that our method explicitly reduces cumulative weight drift from the NC-initialization, compared to standard finetuning. This suggests that our method better preserves the pretrained model's class-specific structure. Empirically, this structural preservation yields Recall@K gains: reduced weight drift correlates with better performance. Concurrent decreases in the Neural Collapse 1 (NC1) measure confirm that stronger intra-class cohesion underlies these improvements. Furthermore, we validate the effectiveness of our method on class-imbalanced benchmarks.



(a) NC-Init with PI (b) Motivational Illustration

Figure 1: (a) *Our Method*: Initializing with collapsed directions leverages transferable structure, while perturbations promote broader utilization of the feature space. (b) *Motivation*: (Top left) Neural collapse leads pretrained features to converge along class-specific directions, resulting in feature collapse on fine-grained data. (Top right) Random initialization misaligns feature representations, thereby underutilizing the transferable structure learned during pretraining. (Bottom right) Our method preserves the transferable structure for coherent downstream clustering.

Introduction

Deep neural classifiers using softmax and cross-entropy loss exhibit a phenomenon called Neural Collapse (NC) (Papyan, Han, and Donoho 2020). Class features converge to their means, forming a simple geometric structure known as an equiangular tight frame (ETF) (Zhu et al. 2021). Moreover, pretrained classifiers exhibit NC, which has been exploited in transfer learning for generalization bounds and transferability (Galanti, György, and Hutter 2022; Li et al. 2022; Wang et al. 2023c). However, despite these promising developments, this emergent geometry remains largely underutilized in downstream tasks such as deep metric learning.

Metric learning (Nie et al. 2023; Yan, Hui, and Li 2023; Wang et al. 2023b) aims to learn an embedding space where similar samples are close and dissimilar ones are far apart. This paradigm has been effectively applied to various tasks, including face recognition (Yi et al. 2014; Liu et al. 2017), image retrieval, and clustering (Movshovitz-Attias et al. 2017; Kim et al. 2020). Among various approaches, proxy-based methods using classification loss offer strong performance in enhanced training stability and Recall@k (Zhai and Wu 2019). This has shown effectiveness on fine-grained domain-specific datasets such as CUB (Wah et al. 2011) and CAR (Krause et al. 2013), which are considered subsets of broader datasets like ImageNet (Deng et al. 2009). Notably, metric learning typically builds on pretrained models trained on large-scale broader datasets, whose final layers exhibit NC. However, this emergent geometry is rarely leveraged in metric learning, which still commonly rely on randomly initialized heads, while prior works have incorporated NC as an inductive bias during training (Zhu et al. 2021; Yang et al. 2022; Kasarla et al. 2022).

To address this oversight, we analyze features extracted from fine-grained datasets using a model pretrained on ImageNet, and report two key observations on how the collapsed

*These authors contributed equally.
†Corresponding author

features of the pretrained model are inherited:

- **Observation 1.** Intra-class Collapse: The embedded features from one class of the fine-grained dataset are already tightly concentrated into its class mean—a direct consequence of NC—even before additional finetuning.
- **Observation 2.** Inter-class Collapse: Due to the nature of domain similarity of fine-grained dataset, the principal component of embeddings of different classes tend to exhibit high angular similarity.

These two observations motivate two key design principles in Fig.1(a). *First,* by initializing the downstream classifier head along the collapsed directions extracted from a pretrained backbone as Observation 1, we fully leverage the rich, transferable structure learned on large-scale data. However, **NC-informed initialization** (NC-Init) alone can leave embeddings confined to a narrow concentrated region, due to Observation 2. *Second,* to address this, we introduce small **Perturbation Injection** (PI) at each update. This perturbation encourages exploration beyond the concentrated regions of the feature space, leading to more generalizable representations (He, Rakin, and Fan 2019; Lim et al. 2021).

We claim that our proposed method preserves transferable structure and induce coherent downstream clustering, as depicted in Fig.1(b). To support this, we empirically confirm that both the drift between initial and final weights (Table 3), indicating preservation of transferable structure, and the NC1 measure (Fig. 5), indicating coherent downstream clustering, are consistently reduced. Moreover, these reductions jointly correlate with improved Recall@k.

Moreover, we theoretically show in Theorem 1 that the hyperparameter of our method offer precise control over cumulative drift during finetuning. Specifically, small perturbations anchor the weights to the NC-initialization, maximizing utilization of the pretrained information, whereas large perturbations allow for broader exploration of the feature space. This interpretable mechanism enables a controllable trade-off between preserving pretrained information and encouraging feature space exploration.

Contributions and Novelty of Our Method

- We propose a novel metric learning method that, for the first time, leverages and preserves the pretrained feature structure by initializing classifier weights along NC directions and introducing controlled perturbations.
- We derive a novel *drift bound*, providing the first theoretical characterization of weight drift in metric learning. Our analysis shows that the proposed perturbation maintains strictly smaller deviation from pretrained weights compared to standard finetuning.
- We empirically show that constrained drift promotes stronger intra-class cohesion, preserving pretrained feature geometry and boosting performance, confirming that maintaining pretrained geometry yields practical improvements in metric learning.

Related Work

Metric Learning for Embedding Space Smoothing. Deep metric learning methods that smooth the embedding space for clustering include techniques based on GANs (Lin et al. 2018), approaches employing multiple proxies with importance scoring (Qian et al. 2019), ensemble methods (Milbich et al. 2020), and probabilistic methods using flow-based models (Roth, Vinyals, and Akata 2022). Augmentation has also been applied through transformation in image space (Fu et al. 2021). Feature mixup (Gu, Ko, and Kim 2021; Venkataramanan et al. 2021; Ko and Gu 2020; Kalantidis et al. 2020) allowed to use the embedding space continuously. In contrast, our method employs perturbation injection to smooth the embedding space, under the guiding principle of preserving as much of the pretrained model's information as possible. Inspired by noise injection (He, Rakin, and Fan 2019; Lim et al. 2021), which injects noise into network training, we apply small isotropic Gaussian perturbations. This NC-Init+Perturbation strategy faithfully preserves the pretrained model's structural priors to smooth decision boundaries and boost metric learning performance.

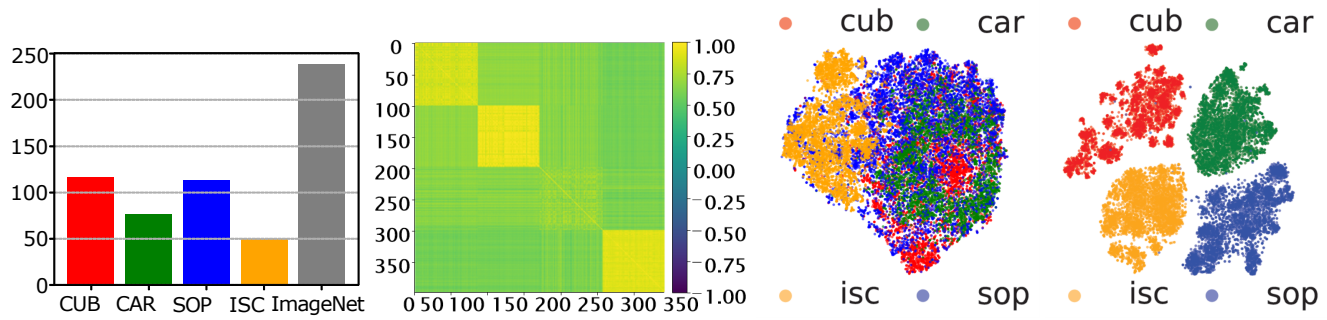
Neural Collapse. The NC phenomenon was first introduced in (Papyan, Han, and Donoho 2020). In (Zhu et al. 2021), a geometric analysis of NC was conducted, and it was studied under normalized conditions in (Yaras et al. 2022), demonstrating that NC also arises in a normalized softmax setting, like metric learning. In (Kasarla et al. 2022), performance improvement and OOD stability were confirmed by incorporating NC as a weight-based inductive bias. More recent studies have extended NC to class-imbalanced learning and analyzed its dynamics under cross-entropy loss (Zhang et al. 2025; Dang et al. 2024), while others have examined NC's role in transfer learning to derive generalization bounds and insights into transferability (Galanti, György, and Hutter 2022; Li et al. 2022; Wang et al. 2023c).

Unlike these methods, our approach introduces a novel weight initialization strategy for metric learning classifiers by leveraging the ETF properties of the pre-trained model.

Weight Initialization. To stabilize and accelerate training, various weight initialization methods (Balduzzi et al. 2017; Zhang, Dauphin, and Ma 2019; Schneider 2022) have been proposed, highlighting their importance. Random initialization of neural networks was introduced in (Glorot and Bengio 2010) and further refined in (He et al. 2015), becoming widely used in computer vision research. As shown in (Aguirre and Fuentes 2019; Das, Bhalgat, and Porikli 2021; Yam et al. 2002; Duch, Adamczak, and Jankowski 1997), if the weights of the neural network represent data clusters, they are well-suited for classification tasks and can accelerate training. In particular, PCA-based initialization of neural networks was proposed in (Seuret et al. 2017; Krähenbühl et al. 2015; Gan et al. 2015). They attempted to initialize weights more uniformly to mitigate vanishing or exploding issues caused by miscalibrated weights.

Neural Collapse-Informed Initialization with Perturbation Injection

We introduce Neural Collapse-Informed Initialization (NC-Init) with Perturbation Injection (PI), a novel method that leverages the structure induced by the NC phenomenon to enhance class representation discriminability and improve



(a) Intra-class variance (b) Class-mean cosine similarity (c) Random-init embeddings (d) Pretrained embeddings

Figure 2: **Characterization of metric-learning benchmark embeddings.** (a) Observation 1: Intra-class variance measured for each dataset and compared against that of ImageNet; (b) Observation 2: Cosine similarity heatmap of the first principal component from the pretrained model (indices 0–98: CAR, 99–198: CUB, 199–298: SOP, 299–398: ISC), showing strong intra-dataset coherence. (c-d) t-SNE produced by a randomly initialized and ImageNet-1K pretrained networks, respectively.

the generalization performance of Deep Metric Learning (DML) models. We first analyze NC in the features of fine-grained datasets produced by a backbone pre-trained on a large-scale, general dataset. We then present our approach in detail and theoretically demonstrate that it better preserves the geometry of the pre-trained embedding.

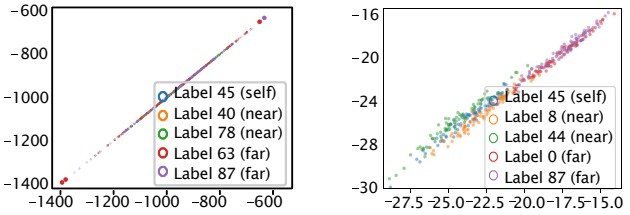
Analysis of the NC Phenomenon

For our analysis in Figs.2 and 3, we used a ResNet-50 with random initialization and ImageNet 1K pretraining. All experiments were executed on an NVIDIA RTX 3090, Intel Xeon 5218, and 128GB RAM with Ubuntu 18.04.

Observation 1: ***Intra-class collapse*** Fig.2(a) shows the trace of average intra-class covariance measured for each dataset compared to ImageNet. Each dataset exhibits lower variance, indicating that the embedded class features in fine-grained datasets are already tightly concentrated around their class centroids before any additional finetuning.

Observation 2: ***Inter-class Collapse*** Fig.2(b) presents a cosine similarity heatmap of the first principal component for each class feature. Fig.2(c)(d) visualizes t-SNE embeddings of a random initialized network and an ImageNet-pretrained network. The visualization confirms that the pretrained backbone already produces dataset-specific clusters before any finetuning, although the clusters are not yet separable at the class level. These findings collectively indicate the presence of inter-class collapse.

This can be explained by the influence of the pretrained model: for example, if it includes a general class such as "cat," and the fine-tuning dataset contains subclasses like "Persian," "Ragdoll," and "Siamese," the resulting embeddings for these subclasses are likely to align along similar directions shaped by the original "cat". To further support this, we include prediction histogram by the ImageNet-pretrained model on the fine-grained datasets in Appendix. We observed that most samples are inferred to a few ImageNet classes that align with their domain.



(a) f_0 (b) f_1

Figure 3: **Analysis on Principal Components Space.** We select a target class and extract features from it and the classes most similar and dissimilar to it. We then project features into the principal component space of the far classes for (a) the random network f_0 and (b) the pretrained network f_1, where f_1 exhibits both intra- and inter-class collapse.

Analysis on Principal Components Space We further investigate these phenomena on the principal components space from feature space of a randomly initialized model (f_0) and and an ImageNet-pretrained model (f_1). We selected a target class (No. 45 in CAR) and defined near and far classes based on cosine similarity between their uncentered first principal components and that of the target class. We construct a projection space using the first principal directions of far classes, expected to capture high between-class variance for analyzing class separability, motivated by Linear Discriminant Analysis. Fig.3 visualizes projections, mirroring the patterns observed in Fig.2 (c)(d): the randomly initialized model (f_0) yields a scattered distribution, while The pretrained model (f_1) exhibits tight clustering with collapsed intra- and inter-class variance. This PCA-based analysis further validates the NC-induced alignment and collapse in the embedding space. Results on the remaining datasets, exhibiting similar trends are in the appendix.

Brief Discussion Through the preceding analyses, we confirmed that, given a domain-specific fine-grained dataset, the pretrained embeddings tend to align and collapse along

specific directions. Since these features inherently reflect the pretrained model's feature structure, we argue that randomly initializing a new classification head and fine-tuning distort the embedded features and degrade the underlying semantic information. To preserve the NC-induced geometry, in the next section, we propose *NC-Init with PI* for maintaining robust, smooth representations that generalize effectively.

Proposed Method

We present our method, **NC-Init+PI**, for metric learning, which explicitly leverages the pretrained neural collapse geometry. **NC-Init** initializes the classifier weights along equiangular tight-frame directions derived from a pretrained model, effectively capturing the rich and transferable structure learned from large-scale data. To prevent overfitting to the narrow region defined by NC-Init and to encourage broader feature exploration, **PI** injects small isotropic Gaussian noise at each update step. We further derive a theoretical drift bound that guarantees limited cumulative weight drift from the NC-initialized parameters, indicating that our method preserves the pretrained structure while enabling robust and discriminative embedding learning.

Notations. We denote our features and labels of training set by $\mathcal{D} = \{(\mathbf{x}, y)\}$, where $\mathbf{x} \in \mathbb{R}^d$ is the d-dimensional embedding, and $y \in \{1, \dots, C\}$ is its class label. The final classification head consists of weights $\{\mathbf{w}_1, \dots, \mathbf{w}_C\} \subset \mathbb{R}^d$, one for each class.We denote by w_c^T the weight after T noisy-gradient updates, and by $w_c^* = \arg\min_w \mathcal{L}_{\mathrm{NS}}(w)$ the global optimum, where $\mathcal{L}_{\mathrm{NS}}$ is the loss defined in Eq.3.

NC-Init (Algorithm 1). We initialize classifier head w_c^0 to the first principal component v_c of its pretrained embedding cluster. This NC-informed initialization anchors the weights to meaningful directions, avoiding the abrupt and arbitrary shifts of features from random initialization, which can erode pretrained semantic information and cause feature degradation.

PI (Algorithm 2). At each training step t, we add isotropic Gaussian noise ε_t to the proxies before computing gradients:

$$\begin{aligned} \widetilde{w}_c^t &= w_c^t + \varepsilon_t, \;\; \varepsilon_t \sim \mathcal{N}(0, \sigma^2 I), \\ w_c^{t+1} &= w_c^t - \eta \, \nabla \mathcal{L}_{\mathrm{NS}}\big(\widetilde{w}_c^t; \{x_i, y_i\}\big), \end{aligned} \tag{1}$$

where $\eta > 0$ is the learning rate used in the gradient update. Although the same σ is used across all classes, the NC-based initialization causes these small perturbations to have a larger relative effect on nearby classes than on distant ones. As a result, the proxies remain anchored to their pretrained directions while still adapting to task-specific adjustments, leading to better preservation of the pretrained structure and improved generalization.

Theoretical Analysis

NC-Init mitigates the abrupt and semantically damaging shifts from random initialization, while PI promotes adaptation to downstream tasks. However, excessive noise in PI may cause weight drift similar to random finetuning, degrading embedding quality. We address this trade-off by deriving a theoretical drift bound, showing that with properly chosen learning rate η and noise scale σ, cumulative deviation from NC initialization is strictly lower than that from random initialization. This bound guarantees that our method preserves the pretrained model's class-specific geometry while permitting controlled feature refinement for downstream tasks.

Algorithm 1: NC-Init

Require: Training set $\mathcal{D}$, classes $\mathcal{C}$, backbone f_1

for $c \in \mathcal{C}$ **do**
  $E_c \leftarrow \{f_1(x) \mid (x, y) \in \mathcal{D},\, y = c\}$
  $v_c \leftarrow \mathrm{PCA}_1(E_c)$
  $w_c^0 \leftarrow v_c / \|v_c\|$
end for
return $\{w_c^0\}$

Algorithm 2: Perturbation Injection

Require: Proxies $\{w_c^t\}$, batch $\{(x_i, y_i)\}$, learning rate η, noise scale σ

for $c \in \mathcal{C}$ **do**
  $\varepsilon_{t,c} \sim \mathcal{N}(0,\, \sigma^2 I)$
  $\widetilde{w}_c^t \leftarrow w_c^t + \varepsilon_{t,c}$
  $\hat{w}_c^t \leftarrow \widetilde{w}_c^t / \|\widetilde{w}_c^t\|$
end for
$\mathcal{L}_t \leftarrow \dfrac{1}{N} \sum_i \mathcal{L}_{\mathrm{NS}}(\{\hat{w}_c^t\}; x_i, y_i)$
for $c \in \mathcal{C}$ **do**
  $w_c^{t+1} \leftarrow w_c^t - \eta \, \nabla_{w_c} \mathcal{L}_t$
end for
return $\{w_c^{t+1}\}$

Definition 1 (Drift from Initialization)**.** To quantify the extent of drift from the initialized proxy v_c, we define the drift D between the v_c and a current weight w_c as the squared ℓ_2 distance $\|w_c - v_c\|^2$. At initialization, $w_c^0 = v_c$, so $D_{\mathrm{init}} = 0$. Under our NC-Init+PI scheme, the drift after T updates is $D_{\mathrm{PI}}(T)$ which incorporates the expectation over the injected Gaussian noise ε. Finally, we define D_{grad} with w_c^* is the global optimum obtained by standard fine-tuning. In summary, this can be compactly written as in

$$\begin{aligned} D_{\mathrm{init}} &:= \|w_c^0 - v_c\|^2 = 0, \\ D_{\mathrm{PI}}(T) &:= \mathbb{E}_\varepsilon\big[\|w_c^T - v_c\|^2\big],\; D_{\mathrm{grad}} := \|w_c^* - v_c\|^2. \end{aligned} \tag{2}$$

Definition 2 (Norm-Softmax loss)**.** The Norm-Softmax loss for a single example $(\mathbf{x}, y)$ is

$$\begin{aligned} \mathcal{L}_{\mathrm{NS}}(w; \mathbf{x}, y) &= -\log \frac{\exp\big((\tilde{\mathbf{x}}^\top \tilde{w}_y)/\tau\big)}{\sum_{c=1}^{C} \exp\big((\tilde{\mathbf{x}}^\top \tilde{w}_c)/\tau\big)}, \\ \text{where} \quad \tilde{\mathbf{x}} &= \frac{\mathbf{x}}{\|\mathbf{x}\|}, \quad \tilde{w}_c = \frac{w_c}{\|w_c\|}, \end{aligned} \tag{3}$$

enforcing unit-norm on both embeddings and proxies, respectively. $\tau > 0$ is the temperature.

Assumption 1 (Regularity Conditions)**.** Let $L > 0$ denote the Lipschitz-gradient constant of $\mathcal{L}_{\mathrm{NS}}$, $\mu > 0$ its PL con-

	BB	Method	CUB				CAR				SOP			
			1	2	4	8	1	2	4	8	1	10	100	1000
ImageNet1K	BN-I	NS_{1024}[1] (Zhai and Wu 2019)	62.2	73.9	82.7	89.4	87.9	93.2	96.2	98.1	74.7	88.3	95.2	-
		MS (Wang et al. 2019)	65.7	77.0	86.3	91.2	84.1	90.4	94.0	96.5	78.2	90.5	96.0	98.7
		SoftTriple (Qian et al. 2019)	65.4	76.4	84.5	90.4	84.5	90.7	94.5	96.9	78.3	90.3	95.9	-
		ProxyGML (Zhu et al. 2020)	66.6	77.6	86.4	-	85.5	91.8	95.3	-	78.0	90.6	96.2	-
		DRML-PA (Zheng et al. 2021)	68.7	78.6	86.3	91.6	86.9	92.1	95.2	97.4	71.5	85.2	93.0	-
		CircleLoss (Sun et al. 2020)	66.7	77.4	86.2	91.2	83.4	89.8	94.1	96.5	78.3	90.5	96.1	98.6
		DAM (Xu et al. 2021)	69.1	79.8	87.2	91.8	86.9	92.1	95.3	97.9	-	-	-	-
		PADS (Roth, Milbich, and Ommer 2020)	66.6	77.2	85.6	-	81.7	88.3	93.0	-	-	-	-	-
		HIST (Lim et al. 2022)	69.7	80.0	87.3	-	87.4	92.5	95.4	-	79.6	91.0	96.2	-
		PA (Kim et al. 2020)	68.4	79.2	86.8	91.6	86.1	91.7	95.0	97.3	79.1	90.8	96.2	98.7
	R50	NS (Zhai and Wu 2019)	61.3	73.9	83.5	90.0	84.2	90.4	94.4	96.9	78.2	90.6	96.2	-
		$Div\&Conq_{128}$ (Sanakoyeu et al. 2019)	65.9	76.6	84.4	90.6	84.6	90.7	94.1	96.5	75.9	88.4	94.9	98.1
		MIC_{128} (Roth, Brattoli, and Ommer 2019)	66.1	76.8	85.6	-	82.6	89.1	93.2	-	77.2	89.4	95.6	-
		$PADS_{128}$ (Roth, Milbich, and Ommer 2020)	67.3	78.0	85.9	-	83.5	89.7	93.8	-	76.5	89.0	95.4	-
		$RankMI_{128}$ (Kemertas et al. 2020)	66.7	77.2	85.1	91.0	83.3	89.8	93.8	96.5	74.3	87.9	94.9	98.3
		EPSHN (Xuan, Stylianou, and Pless 2020)	64.9	75.3	83.5	-	82.7	89.3	93.0	-	78.3	90.7	96.3	-
		DiVA (Milbich et al. 2020)	69.2	79.3	-	-	87.6	92.9	-	-	79.6	91.2	-	-
		IBC (Seidenschwarz, Elezi, and Leal-Taixé 2021)	70.3	80.3	87.6	92.7	88.1	93.3	96.2	**98.2**	81.4	91.3	95.9	-
		HIST (Lim et al. 2022)	71.4	81.1	88.1	-	**89.6**	**93.9**	**96.4**	-	81.4	92.0	96.7	-
		HIER (Kim, Jeong, and Kwak 2023)	70.1	79.4	86.9	-	88.2	93.0	95.6	-	80.2	91.5	96.6	-
		PA (Kim et al. 2020)	69.7	80.0	87.0	92.4	87.7	92.9	95.8	97.9	-	-	-	-
ImageNet21K	R50	PA (reproduced)	81.8	88.5	92.9	95.7	86.8	92.4	95.5	97.5	81.8	92.6	97.1	99.0
		PA + Ours	**83.2**	**89.2**	**93.3**	**95.8**	89.2	93.4	96.2	97.8	**82.1**	**92.7**	**97.2**	**99.1**
	ViT	HIER (Kim, Jeong, and Kwak 2023)	85.7	91.3	94.4	-	88.3	93.2	96.1	-	86.1	95.0	98.0	-
		VPTSP-G [2] (Ren et al. 2024)	**86.6**	**91.7**	**94.8**	-	87.7	93.3	96.1	-	84.4	93.6	97.3	-
		DFML-PA (Wang et al. 2023a)	79.1	86.8	-	-	**89.5**	93.9	-	-	84.2	93.8	-	-
		HypViT (Ermolov et al. 2022)	85.6	91.4	**94.8**	96.7	86.5	92.1	95.3	97.3	85.9	94.9	98.1	**99.5**
		HypViT + Ours	85.9	91.5	**94.8**	**96.8**	88.5	93.2	95.7	97.6	**86.3**	**95.2**	**98.2**	**99.5**

Table 1: **Comparison on the CUB, CAR, and SOP datasets** in Recall@k. Pre-trained datasets are grouped first, followed by backbones and their respective methods. The best performance is boldfaced, while the second-best performance is underlined.

stant, and $G > 0$ an upper bound on its gradient norm.

$$\|\nabla\mathcal{L}(u) - \nabla\mathcal{L}(v)\| \le L\,\|u - v\|, \tag{A1}$$

$$\tfrac{1}{2}\,\|\nabla\mathcal{L}(w)\|^2 \ge \mu\,\big(\mathcal{L}(w) - \mathcal{L}(v)\big), \tag{A2}$$

$$\|\nabla\mathcal{L}_{\mathrm{NS}}(w)\| \le G. \tag{A3}$$

Theorem 1 (T-step Drift Upper Bound)**.** *Under Assumption 1, for any time step $T \ge 1$, $\exists(\eta, \sigma)$ s.t. our PI drift satisfies:*

$$D_{\mathrm{PI}}(T) \;\le\; \frac{\eta^2 G^2 + \eta\frac{L^2 d\,\sigma^2}{\mu}}{\eta\mu}\Big(1 - (1-\eta\mu)^T\Big). \tag{5}$$

$$\limsup_{T\to\infty} D_{\mathrm{PI}}(T) \;\le\; R^2 < D_{\mathrm{grad}},$$
$$\textit{where}\quad R^2 = \frac{\eta\,G^2}{\mu} + \frac{L^2 d\,\sigma^2}{\mu^2}. \tag{6}$$

This suggest that despite the perturbations ($\sigma > 0$), NC-Init+PI preserves the pretrained proxy directions more faithfully than pure-gradient training. Although our theoretical analysis focuses on classifier weights, the alignment between features and weights in the terminal phase of training suggests that similar conclusions hold for the feature representations.

[1]Subscript specifies the embedding dimension (default 512).

[2]It leverages LLM model.

	BB	Method	ISC			
			1	10	20	30
ImageNet1K	BN-I	NS_{1024}	86.6	97.0	98.0	98.5
		MS	89.7	97.9	98.5	98.8
		PA	91.5	98.1	98.8	–
	R50	NS	88.6	97.5	98.4	98.8
		$Div\&Conq_{128}$	85.7	95.5	96.9	97.5
		MIC_{128}	88.2	97.0	–	98.0
		EPSHN	87.8	95.7	96.8	–
		IBC	**92.8**	**98.5**	**99.1**	–
		HIER	92.4	98.2	98.8	–
ImageNet21K	R50	PA (reproduced)	92.0	98.2	98.8	99.1
		PA + Ours	92.5	98.2	98.8	99.1
	ViT	HIER	92.8	**98.4**	**99.0**	–
		VPTSP-G	91.2	97.6	98.4	–
		HypViT	92.5	98.3	98.8	99.1
		HypViT + Ours	**92.9**	**98.4**	**99.0**	**99.2**

Table 2: **Comparison on the ISC dataset** in Recall@k.

Experiments

We followed the standard protocol in DML (Oh Song et al. 2016) to evaluate the effectiveness of the proposed method. The experiments were conducted on widely used metric learning benchmark datasets, including Caltech-UCSD Birds (CUB) (Wah et al. 2011), CARS196 (CAR) (Krause et al. 2013), Stanford Online Products (SOP)(Oh Song et al. 2016), and InShop Cloths (ISC) (Liu et al. 2016). We selected several baseline methods and evaluated the proposed NC-init and PI techniques by applying them to these base-

CUB						
η	Baseline		NC-Init only		NC-Init+PI	
	Drift	R@1	Drift	R@1	Drift	R@1
1×10^{-4}	2.0106	81.18	0.1214	80.33	0.0878	80.55
5×10^{-4}	2.0154	81.63	0.3218	82.14	0.3257	82.34
1×10^{-3}	2.0146	81.75	0.5146	82.70	0.5512	83.23
5×10^{-3}	2.0210	81.85	1.1606	82.48	1.1609	82.48
CAR						
η	Baseline		NC-Init only		NC-Init+PI	
	Drift	R@1	Drift	R@1	Drift	R@1
1×10^{-4}	2.0047	81.68	0.1620	80.25	0.1615	80.62
5×10^{-4}	2.0112	83.22	0.3490	84.81	0.3496	84.90
1×10^{-3}	2.0164	84.28	0.5500	86.11	0.5502	86.34
5×10^{-3}	2.0275	86.80	1.1896	88.70	1.1912	89.15

Table 3: **Comparison of three configurations of our method**—baseline, NC-Init only, and NC-Init+PI—across different warm-up LR η on the CUB and CAR datasets.

lines. Specifically, we chose Proxy Anchor (CNN-based) and HypViT (ViT-based) as baselines due to their relatively few hyperparameters and strong reproducibility, making them well-suited for evaluation.

To ensure a fair comparison, all baseline hyperparameters, including the optimizer, embedding size, and pooling type, were kept consistent across experiments. For Proxy Anchor, all experimental settings remained identical to the baseline, except for the hyperparameters (η, σ) introduced in the proposed method. In contrast, HypViT required additional modifications. Since the baseline lacks a pre-softmax layer corresponding to class weights, we added a head to the ViT model to incorporate class weights initialized by our proposed NC-Init. Additionally, we introduced a NormSoftmax loss with perturbations applied to the class weights, complementing HypViT's original pairwise cross-entropy loss to effectively learn class weights during training. To demonstrate how effectively our proposed method leverages the pre-trained dataset, we also compared Proxy Anchor with a ResNet50 backbone pre-trained on ImageNet-21K. By analyzing the difference between the reproduced baseline results and the outcomes of our proposed method in this setting, we highlight the effectiveness of our approach in utilizing the advantages of pre-trained models.

The dataset-specific hyperparameters (warmup learning rate η, perturbation noise σ) were set as follows: for CUB, (η, σ) = (0.001, 0.01); for CAR, (0.005, 0.0001); for SOP, (0.005, 0.0001), and for ISC, (0.005, 0.001).

Comparison with Other DML Methods

Tables 1 and 2 compare the proposed method with other state-of-the-art approaches across the CUB, CAR, SOP, and ISC datasets. Table 1 focuses on CNN-based and ViT-based methods evaluated on the CUB, CAR, and SOP datasets, while Table 2 highlights results of the ISC dataset. To facilitate comparison, methods were grouped into CNN-based and ViT-based categories. Within each group, the best-performing result was highlighted in **boldface** and the second-best result in underlined.

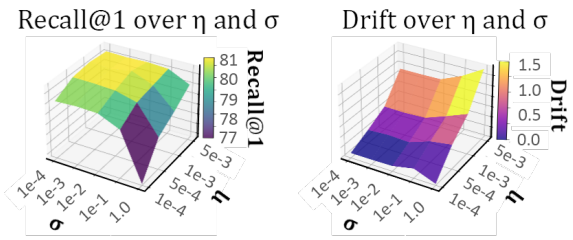


Figure 4: **Recall@1 vs. Drift** over the grid of warm-up LR η and PI noise levels σ on CUB datset.

In the CUB dataset, our method, PA + Ours, combined with PA and ImageNet-21K pre-training, achieved the highest performance among CNN-based methods. It surpasses the second-best method, HIST, by a substantial margin of 11.8 pts in Recall@1 (83.2% vs. 71.4%). While the use of ImageNet-21K as a pre-training dataset significantly contributed to the performance gains, the additional improvement of 1.4% over the baseline PA (PA (reproduced)) shows that our method not only benefits from a larger pre-trained dataset but also effectively enhances generalization. For ViT-based methods, the proposed HypViT + Ours achieved the second-highest performance, with Recall@1 of 85.9. The best performance was recorded by VPTSP-G (86.6), a method that uses prompts. In contrast to the added complexity and information of VPTSP-G, our approach maintains a simpler architecture and delivers competitive performance.

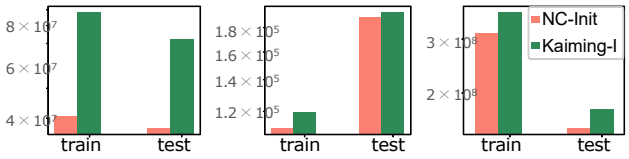
When the proposed method is applied to PA trained on ImageNet-21K, performance increases by 2.4% (from 86.8 to 89.2 in Recall@1), resulting in the second-highest score among CNN-based methods. Additionally, for HypViT + Ours, a 2% improvement in Recall@1 was observed on the CAR dataset, increasing from 86.5 to 88.5 compared to the baseline HypViT. The substantial performance boost in CNN backbones indicates that *the proposed method effectively mitigates the limitations of using a larger pre-trained dataset, possibly by improving alignment and regularization effects.*

On SOP, our method boosted Recall@1 from 81.8 to 82.1 (+0.3%) for PA and from 85.9 to 86.3 (+0.4%) for HypViT, achieving the best results in each category. On ISC, it improved Recall@1 by 0.5% (92.0→92.5) for PA and by 0.4% (92.5→92.9) for HypViT, ranking second among CNNs and first among ViTs—demonstrating consistent effectiveness across architectures.

Ablation Study on the Proposed Method

The proposed method begins with a *warmup* stage, during which the pretrained backbone is frozen—except for its batch normalization layers—and only the newly added embedding projection and proxy weights are trained using a learning rate η and PI noise σ. After the warmup phase, the entire network is unfrozen and trained using the baseline learning rate schedule.

In this ablation study, we varied the *warmup* learning rate (warmup LR) η and the PI noise σ, while keeping the main-



(a) Intra-class (b) Inter-class (c) NC1

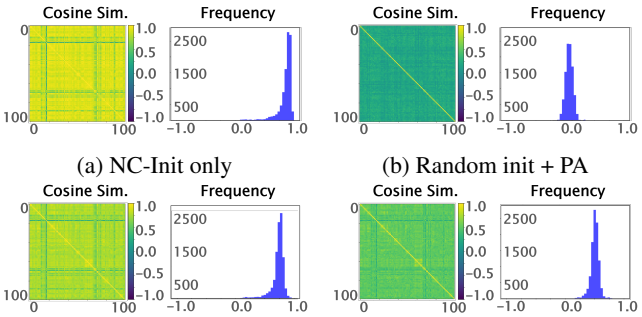
Figure 5: **Intra-/inter-class variances and NC1** for CUB.

training schedule fixed. We performed a sweep over the following hyperparameters: $\eta \in \{0.0001,\ 0.0005,\ 0.001,\ 0.005\}$, $\sigma \in \{0,\ 0.0001,\ 0.001,\ 0.01,\ 0.1,\ 1.0\}$ and evaluated each configuration by measuring the final proxy drift $D_{\mathrm{PI}}(T)$ in Eq.2 after the warmup stage, along with the Recall@1 on the CUB and CAR datasets. In addition, we evaluated the contributions of each component: NC-Init and PI.

Analysis on drift and clustering. Table 3 presents the proxy drift and Recall@1 for each warmup LR η, evaluated under three configurations: the random-initialization baseline (no PI), NC-Init only ($\sigma = 0$), and NC-Init + PI with $\sigma = 10^{-2}$ on CUB and $\sigma = 10^{-4}$ on CAR. Once NC-Init is applied with a suitably chosen warm-up η, proxy drift sharply decreases from approximately 2.0 to within the range [0.1, 1.2]. This result empirically supports our theoretical claim that NC-Init aligns proxies closely with their pretrained NC directions. Adding PI on top of NC-Init yields drift values comparable to NC-Init alone—except for a slight increase at the smallest η—while consistently improving Recall@1 for all but the smallest η setting on both datasets. These findings suggest that PI offers a modest yet consistent enhancement in retrieval performance without significantly disrupting the NC alignment preserved by NC-Init.

Hyperparameter vs. performance. Fig.4 presents the complete surfaces of Recall@1 and drift over the $(\eta, \sigma > 0)$ grid. The performance surface indicates that Recall@1 is maximized when moderate to large warm-up learning rates are paired with small noise magnitudes. The corresponding drift surface shows that this performance peak coincides with an intermediate level of drift—neither minimal nor excessive. These findings support the conclusion that optimal performance in metric learning arises from a balance between preserving pretrained geometric structure and allowing sufficient adaptation. This balance can be effectively tuned via the warm-up learning rate and perturbation.

Analysis on intra/inter covariance. Fig.5 reports the trace of intra-class covariance, inter-class covariance, and the NC1 metric (capturing the relative intra-class variance normalized by inter-class variance (Papyan, Han, and Donoho 2020)) for CUB. Compared to a random-init baseline, NC-Init+Perturbation achieves a modest reduction in inter-class variance but, crucially, a much larger decrease in intra-class variance, indicating tighter class cohesion. Consequently, the NC1 for our method are lower, demonstrating enhanced class separability with tighter class cohesion.



(a) NC-Init only (b) Random init + PA

(c) NC-Init + PA (d) NC-Init + PI + PA

Figure 6: **Proxy-to-proxy affinity matrices and off-diagonal cosine similarity histograms**. (a) immediately after NC-Init (R@1=62.20); (b) after random-Init + PA (R@1=81.75); (c) after NC-Init + PA (R@1=82.70); (d) after NC-Init + PI + PA (R@1=83.23).

Effects: Classifier-Weight Analysis

Fig.6 displays the proxy-to-proxy cosine similarity and corresponding off-diagonal similarity density across four distinct training procedures. In (a), we show the state immediately following NC-Init(prior to any metric-learning updates), where the cosine similarity appears almost entirely yellow, and the off-diagonal histogram peaks near 1. This reflects the strong NC structure present in the pretrained embeddings. Fig.6(b) illustrates the result of applying the Proxy-Anchor loss to a randomly initialized classifier, where inter-class similarities shrink toward zero, with off-diagonal similarities concentrated in the 0–0.3 range. In (c), NC-Init followed by Proxy-Anchor training reduces the initially high affinities seen in the pretrained state, while still preserving moderate inter-class alignment compared to random initialization. Finally, Fig.6(d) presents the full NC-Init + PI + Proxy-Anchor pipeline: perturbation injection further suppresses inter-proxy similarities, yielding affinity values lower than in (c) but higher than the near-zero levels of (b). This progression illustrates how our method effectively balances the retention of pretrained structure with the flexibility required for fine-grained adaptation.

Conclusion

In this paper, we introduce a simple yet principled method that first aligns classifier weights with the NC directions of a pretrained model, followed by the injection of small isotropic Gaussian noise. We theoretically demonstrate that this NC-Init + Perturbation scheme explicitly constrains cumulative weight drift from the pretrained solution via a novel drift bound, ensuring strictly smaller deviations compared to conventional fine-tuning. Empirically, we show that moderate, bounded drift consistently correlates with the highest Recall@K, confirming that our theoretical insights yield tangible retrieval improvements. To our knowledge, we are the first to propose that preserving Neural Collapse structure benefits fine-tuning; further research is needed to elucidate the information encoded by pretrained models and to develop strategies for exploiting it in fine-grained tasks.

Acknowledgments

We are deeply grateful to Inwon Lee for her invaluable support and encouragement. We are also grateful to Won-Seok Choi for insightful scholarly discussions. This work was partly supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) (RS-2024-00456709/34%, RS-2021-II211341-Artificial Intelligence Graduate School Program(Chung-Ang University)/33%, RS-2021-II211343-GSAI/10%, RS-2022-II220953-PICA/5%, RS-2021-II212068-AIHub/3%, RS-2022-II220951-LBA/3%), NRF (RS-2024-00353991-SPARC/3%, RS-2023-00274280-HEI/3%), KEIT (RS-2024-00423940/3%), and Gwangju Metropolitan City (Artificial intelligence industrial convergence cluster development project/3%) grant funded by the Korean government.

References

Aguirre, D.; and Fuentes, O. 2019. Improving Weight Initialization of ReLU and Output Layers. In *ICANN*.

Balduzzi, D.; Frean, M.; Leary, L.; Lewis, J.; Ma, K. W.-D.; and McWilliams, B. 2017. The shattered gradients problem: If resnets are the answer, then what is the question? In *ICML*.

Dang, H.; Tran, T.; Nguyen, T.; and Ho, N. 2024. Neural collapse for cross-entropy class-imbalanced learning with unconstrained relu feature model. *arXiv preprint arXiv:2401.02058*.

Das, D.; Bhalgat, Y.; and Porikli, F. 2021. Data-driven Weight Initialization with Sylvester Solvers. *arXiv preprint arXiv:2105.10335*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*.

Duch, W.; Adamczak, R.; and Jankowski, N. 1997. Initialization and optimization of multilayered perceptrons. In *NEURAP*, 99–104.

Ermolov, A.; Mirvakhabova, L.; Khrulkov, V.; Sebe, N.; and Oseledets, I. 2022. Hyperbolic vision transformers: Combining improvements in metric learning. In *CVPR*.

Fu, Z.; Li, Y.; Mao, Z.; Wang, Q.; and Zhang, Y. 2021. Deep metric learning with self-supervised ranking. In *AAAI*.

Galanti, T.; György, A.; and Hutter, M. 2022. Improved generalization bounds for transfer learning via neural collapse. In *ICML Workshop*.

Gan, Y.; Liu, J.; Dong, J.; and Zhong, G. 2015. A PCA-based convolutional network. *arXiv preprint arXiv:1505.03703*.

Glorot, X.; and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTAT*.

Gu, G.; Ko, B.; and Kim, H.-G. 2021. Proxy synthesis: Learning with synthetic classes for deep metric learning. In *AAAI*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv preprint arXiv:1502.01852*.

He, Z.; Rakin, A. S.; and Fan, D. 2019. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *CVPR*.

Kalantidis, Y.; Sariyildiz, M. B.; Pion, N.; Weinzaepfel, P.; and Larlus, D. 2020. Hard negative mixing for contrastive learning. In *NIPS*, 21798–21809.

Kasarla, T.; Burghouts, G.; van Spengler, M.; van der Pol, E.; Cucchiara, R.; and Mettes, P. 2022. Maximum class separation as inductive bias in one matrix. In *NeurIPS*.

Kemertas, M.; Pishdad, L.; Derpanis, K. G.; and Fazly, A. 2020. Rankmi: A mutual information maximizing ranking loss. In *CVPR*.

Kim, S.; Jeong, B.; and Kwak, S. 2023. HIER: Metric Learning Beyond Class Labels via Hierarchical Regularization. In *CVPR*.

Kim, S.; Kim, D.; Cho, M.; and Kwak, S. 2020. Proxy anchor loss for deep metric learning. In *CVPR*.

Ko, B.; and Gu, G. 2020. Embedding expansion: Augmentation in embedding space for deep metric learning. In *CVPR*, 7255–7264.

Krähenbühl, P.; Doersch, C.; Donahue, J.; and Darrell, T. 2015. Data-dependent initializations of convolutional neural networks. *arXiv preprint arXiv:1511.06856*.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *ICCV Workshops*.

Li, X.; Liu, S.; Zhou, J.; Lu, X.; Fernandez-Granda, C.; Zhu, Z.; and Qu, Q. 2022. Understanding and improving transfer learning of deep models via neural collapse. *arXiv preprint arXiv:2212.12206*.

Lim, J.; Yun, S.; Park, S.; and Choi, J. Y. 2022. Hypergraph-induced semantic tuplet loss for deep metric learning. In *CVPR*.

Lim, S. H.; Erichson, N. B.; Utrera, F.; Xu, W.; and Mahoney, M. W. 2021. Noisy Feature Mixup. *arXiv preprint arXiv:2110.02180*.

Lin, X.; Duan, Y.; Dong, Q.; Lu, J.; and Zhou, J. 2018. Deep variational metric learning. In *ECCV*.

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L. 2017. SphereFace: Deep Hypersphere Embedding for Face Recognition. In *CVPR*.

Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; and Tang, X. 2016. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*.

Milbich, T.; Roth, K.; Bharadhwaj, H.; Sinha, S.; Bengio, Y.; Ommer, B.; and Cohen, J. P. 2020. Diva: Diverse visual feature aggregation for deep metric learning. In *ECCV*.

Movshovitz-Attias, Y.; Toshev, A.; Leung, T. K.; Ioffe, S.; and Singh, S. 2017. No fuss distance metric learning using proxies. In *ICCV*.

Nie, J.; Dong, Z.; He, Z.; Wu, H.; and Gao, M. 2023. FAML-RT: Feature alignment-based multi-level similarity metric learning network for a two-stage robust tracker. *Information Sciences*, 632: 529–542.

Oh Song, H.; Xiang, Y.; Jegelka, S.; and Savarese, S. 2016. Deep metric learning via lifted structured feature embedding. In *CVPR*.

Papyan, V.; Han, X.; and Donoho, D. L. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663.

Qian, Q.; Shang, L.; Sun, B.; Hu, J.; Li, H.; and Jin, R. 2019. Softtriple loss: Deep metric learning without triplet sampling. In *ICCV*.

Ren, L.; Chen, C.; Wang, L.; and Hua, K. 2024. Learning Semantic Proxies from Visual Prompts for Parameter-Efficient Fine-Tuning in Deep Metric Learning. *arXiv preprint arXiv:2402.02340*.

Roth, K.; Brattoli, B.; and Ommer, B. 2019. Mic: Mining interclass characteristics for improved metric learning. In *ICCV*.

Roth, K.; Milbich, T.; and Ommer, B. 2020. Pads: Policy-adapted sampling for visual similarity learning. In *CVPR*.

Roth, K.; Vinyals, O.; and Akata, Z. 2022. Non-isotropy regularization for proxy-based deep metric learning. In *CVPR*.

Sanakoyeu, A.; Tschernezki, V.; Buchler, U.; and Ommer, B. 2019. Divide and conquer the embedding space for metric learning. In *CVPR*.

Schneider, J. 2022. Correlated Initialization for Correlated Data. *Neural Processing Letters*, 54(3): 2249–2266.

Seidenschwarz, J. D.; Elezi, I.; and Leal-Taixé, L. 2021. Learning intra-batch connections for deep metric learning. In *ICML*.

Seuret, M.; Alberti, M.; Liwicki, M.; and Ingold, R. 2017. PCA-Initialized Deep Neural Networks Applied to Document Image Analysis. In *ICDAR*.

Sun, Y.; Cheng, C.; Zhang, Y.; Zhang, C.; Zheng, L.; Wang, Z.; and Wei, Y. 2020. Circle loss: A unified perspective of pair similarity optimization. In *CVPR*.

Venkataramanan, S.; Psomas, B.; Kijak, E.; Amsaleg, L.; Karantzalos, K.; and Avrithis, Y. 2021. It takes two to tango: Mixup for deep metric learning. *arXiv preprint arXiv:2106.04990*.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset.

Wang, C.; Zheng, W.; Li, J.; Zhou, J.; and Lu, J. 2023a. Deep factorized metric learning. In *CVPR*.

Wang, S.; Zeng, Q.; Zhang, X.; Ni, W.; and Cheng, C. 2023b. Multi-modal pseudo-information guided unsupervised deep metric learning for agricultural pest images. *Information Sciences*, 630: 443–462.

Wang, X.; Han, X.; Huang, W.; Dong, D.; and Scott, M. R. 2019. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*.

Wang, Z.; Luo, Y.; Zheng, L.; Huang, Z.; and Baktashmotlagh, M. 2023c. How far pre-trained models are from neural collapse on the target dataset informs their transferability. In *ICCV*, 5549–5558.

Xu, F.; Wang, M.; Zhang, W.; Cheng, Y.; and Chu, W. 2021. Discrimination-aware mechanism for fine-grained representation learning. In *CVPR*.

Xuan, H.; Stylianou, A.; and Pless, R. 2020. Improved embeddings with easy positive triplet mining. In *WACV*.

Yam, Y.-F.; Leung, C.-T.; Tam, P. K.; and Siu, W.-C. 2002. An independent component analysis based weight initialization method for multilayer perceptrons. *Neurocomputing*, 48(1-4): 807–818.

Yan, M.; Hui, S. C.; and Li, N. 2023. DML-PL: Deep metric learning based pseudo-labeling framework for class imbalanced semi-supervised learning. *Information Sciences*, 626: 641–657.

Yang, Y.; Chen, S.; Li, X.; Xie, L.; Lin, Z.; and Tao, D. 2022. Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network? In *NeurIPS*, 37991–38002.

Yaras, C.; Wang, P.; Zhu, Z.; Balzano, L.; and Qu, Q. 2022. Neural collapse with normalized features: A geometric analysis over the riemannian manifold. *arXiv preprint arXiv:2209.09211*.

Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z. 2014. Deep metric learning for person re-identification. In *ICPR*.

Zhai, A.; and Wu, H.-Y. 2019. Classification is a Strong Baseline for Deep Metric Learning. *arXiv preprint arXiv:1811.12649*.

Zhang, E.; Li, C.; Geng, C.; and Chen, S. 2025. All-around neural collapse for imbalanced classification. *IEEE Transactions on Knowledge and Data Engineering*.

Zhang, H.; Dauphin, Y. N.; and Ma, T. 2019. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*.

Zheng, W.; Zhang, B.; Lu, J.; and Zhou, J. 2021. Deep relational metric learning. In *ICCV*.

Zhu, Y.; Yang, M.; Deng, C.; and Liu, W. 2020. Fewer is more: A deep graph metric learning perspective using fewer proxies. In *NeurIPS*.

Zhu, Z.; Ding, T.; Zhou, J.; Li, X.; You, C.; Sulam, J.; and Qu, Q. 2021. A geometric analysis of neural collapse with unconstrained features. In *NeurIPS*.