南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Inducing Neural Collapse to a Fixed Hierarchy-Aware Frame for Reducing Mistake Severity

Tong Liang, Jim Davis
Ohio State University
Columbus, Ohio 43210
{liang.693, davis.1719}@osu.edu

ICCV 2023

Background

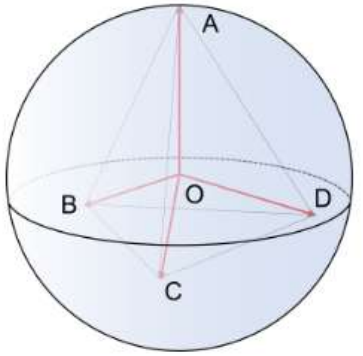
Neural Collapse: The final layer classifier in a deep neural network tends to have a simple symmetric structure (simplex Equiangular Tight Frame, ETF) :

(NC1) Cross-example within-class variability of last-layer training activations collapsesto zero, as the individual features collapse to their class means.

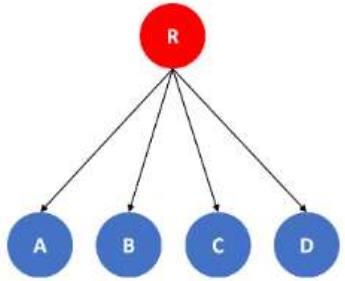
(NC2) The class means collapse to ETF.

(NC3) Self-duality, the classifier vector also presents ETF, and the classifier matrix and feature matrix are linearly related: $\frac{W}{\|W\|_F} = \frac{H}{\|H\|_F}$

(NC4) The network's decision collapses to simply choosing the class with the closest Euclidean distance between its class mean and features of the test example.



(a)



(c)

$$\mathbf{v} \in \mathbb{R}^d,\ d \geq K-1$$

$$\cos(\boldsymbol{v}_i, \boldsymbol{v}_j) = -\frac{1}{K-1}$$

Prevalence of Neural Collapse During the Terminal Phase of Deep Learning Training. Proceedings of the National Academy of Sciences of the United States of America (PNAS), 117, 09 2020.

Introduction

Neural collapse phenomenon makes sense considering an ETF separates all classes equally and maximally from each other.

However, such a structure may not emerge when trained with imbalanced data. Features of minor classes may collapse to the same vector (minority collapse)

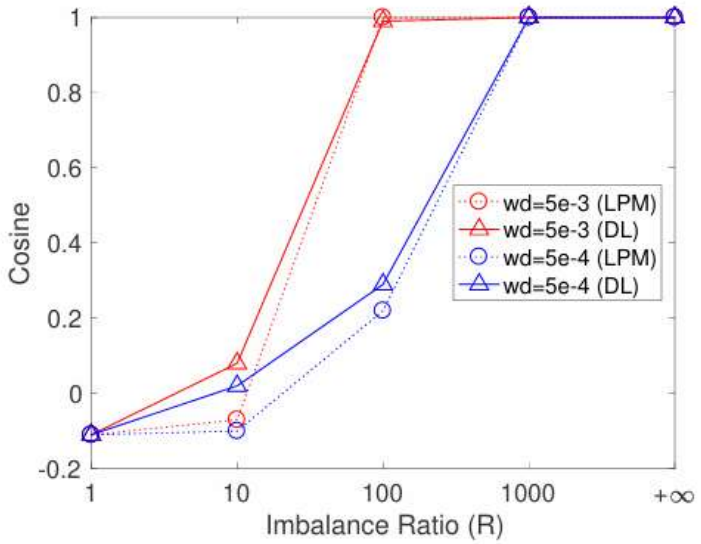


Fig. 2. Minority Collapse predicted by the Layer-Peeled Model (LPM, in dotted lines) and empirically observed in deep learning (DL, in solid lines) on imbalanced datasets with $K_A = 7$ and $K_B = 3$. The y-axis denotes the average cosine of the angles between any pair of the minority classifier $\boldsymbol{w}^\star_{K_A+1}, \ldots, \boldsymbol{w}^\star_K$ for both LPM and DL. The datasets we use are subsets of the CIFAR10 datasets (12) and the size of the majority classes is fixed to 5000. The experiments use VGG13 (13) as the deep learning architecture, with weight decay (wd) $\lambda = 5 \times 10^{-3}, 5 \times 10^{-4}$. The prediction is especially accurate in capturing the phase transition point where the cosine becomes 1 or, equivalently, the minority classifiers become parallel to each other. More details can be found in Section C.

Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. Proceedings of the National Academy of Sciences of the United States of America (PNAS), 2021.

Introduction

Related works:

- Encourage the features to form an ETF structure by fixing the classifier weights at a pre-computed ETF.
- Employing additional regularizes to induce neural collapse.

Concerns:

- When the ETF classifier makes a mistake, it is mainly random due to its equiangular nature.
- Conventional neural networks are trained mainly with cross-entropy and one-hot labels, ignoring any underlying hierarchical label relationships.

In real-world application scenarios, some classification mistakes would have a much worse impact than others. It is critical to incorporate mistake severity into the performance evaluations.

Related works

Hierarchy-aware cost methods

$$\operatorname{argmin}_k R(\mathbf{y}=k|\mathbf{x}) = \operatorname{argmin}_k \sum_{j=1}^{K} \mathbf{C}_{k,j} \cdot p(\mathbf{y}=j|\mathbf{x})$$

No Cost Likelihood Manipulation at Test Time for Making Better Mistakes in Deep Networks. ICLR, 2021.

Hierarchy-aware architecture methods

the authors proposed to use multiple classification heads for different levels of classes in the hierarchy, where the penultimate features are decoupled into different segments for the respective coarse and fine-grained classifiers.

Flamingo" is My "Bird": Fine-Grained, or Not. CVPR, 2021.

Hierarchy-aware label methods

Using Soft-labels. The classifiers of all levels share the same penultimate features from the backbone network. Enforce by minimizing the JS Divergence between predictions of the coarse-level classifier and the soft-labels reconstructed from predictions of the next fine-level classifier.

Learning Hierarchy Aware Features for Reducing Mistake Severity. ECCV, 2022.

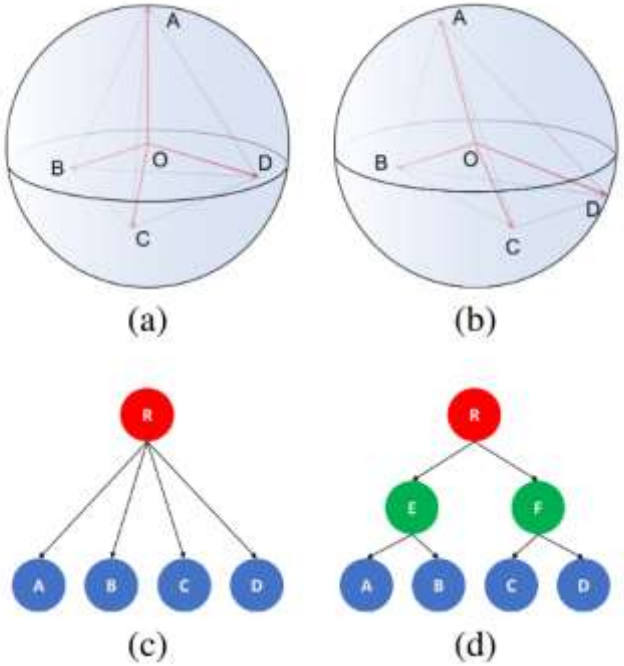
Method

Inspiration: simplex ETF

Compared to the ETF for four classes, the HAFrame captures the pair-wise hierarchical distances across the four classes from a hierarchy shown in Fig. 1(d), with class A closer to B, and A equally distant to C and D.

Purpose: fix the linear classifier to a HAFrame with the hierarchical relationship and induce features to HAFrame.

Training strategy: employ a weighted loss consisting of the cross-entropy loss and the proposed cosine similarity-based auxiliary loss to induce penultimate features collapsing onto the associated classifier vectors (HAFrame).



Method

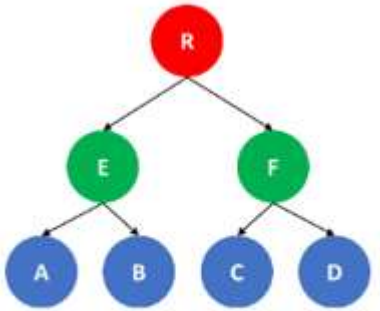
Pair-wise Cosine Similarity

$$d_{i,j} = \text{height}(\text{LCA}(y_i, y_j))$$

$$S_{ij} = (1 - s_{\min}) \cdot e^{-\gamma \cdot \frac{d_{ij}}{d_{\max}}} + s_{\min}$$

$s_{\min} \in (-1,1)$

A larger γ makes the hierarchical distance have a greater impact on the similarity.



(d)

Similarity Hierarchy-Aware Frame

$$\mathbf{S} = \mathbf{W}^T\mathbf{W}$$

$$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K], \mathbf{w}_i \in \mathbb{R}^K, ||\mathbf{w}_i||_2 = 1$$

$$cos\angle(\boldsymbol{w}_i, \boldsymbol{w}_j) = \boldsymbol{w}_i^T \boldsymbol{w}_j = S_{ij}, \forall 1 \le i \le j \le K$$

Method

we can find **W** by the following matrix factorization:

$$\mathbf{S} = \mathbf{W}^T\mathbf{W}$$
$$= \mathbf{Q}\mathbf{D}\mathbf{Q}^T \text{ (eigenvalue decomposition)}$$

$$= \left(\mathbf{Q}\mathbf{D}^{1/2}\mathbf{U}^T\right)\left(\mathbf{U}\mathbf{D}^{1/2}\mathbf{Q}^T\right) = \mathbf{W}^T\mathbf{W}$$

$$\begin{bmatrix} a & 1-a \\ b & 1-b \end{bmatrix} \text{ -> } \begin{bmatrix} a & 1-a \\ 0 & 1-\frac{b}{a} \end{bmatrix} \text{ (a>b)}$$

$\mathbf{U} \in \mathbb{R}^{K\times K}$ is an orthonormal matrix obtained from QR-decomposition of a randomly sampled matrix in $\mathbb{R}^{K\times K}$

Method

Additional Transformation Layer

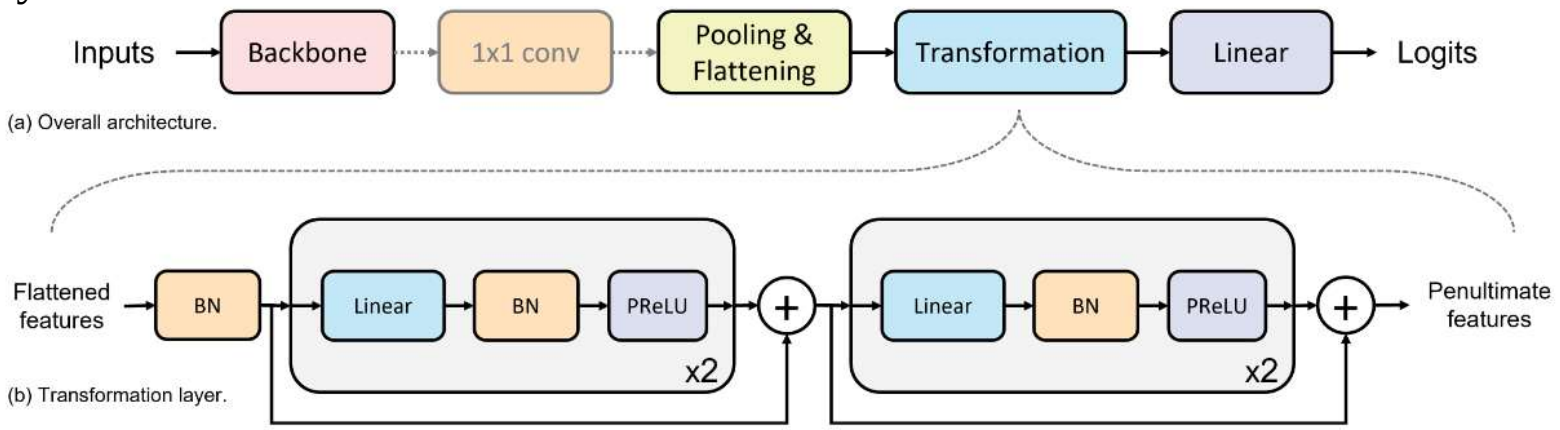


Figure 3. Illustration of the customized network architectures. (a) Top: overall network architecture of our approach, the 1x1 convolutional layer is only used in our type-II models. (b) Bottom: the proposed transformation layer, where BN is 1D batch norm layer.

Have an inductive bias towards non-negative values due to the use of nonlinear activation functions (e.g., ReLU, GELU. etc.).

Our proposed transformation layer uses parametric ReLU (PReLU), which learns the slope of the rectified linear function for negative inputs to mitigate the aforementioned bias

$$\mathrm{PReLU} = \begin{cases} x, & \text{if} x \geq 0 \\ ax, & \text{otherwise} \end{cases}$$

Method

Cosine Similarity Based Auxiliary Loss

Logits:

$$\hat{y} = \arg\max_i \boldsymbol{W}^T \boldsymbol{h} = \arg\max_i cos\angle(\boldsymbol{w}_i, \boldsymbol{h})$$

Facilitate the collapse of penultimate features onto the respective classifier weights:

$$\mathcal{L}_{COS} = \sum_{i=1}^{K} (cos\angle(\boldsymbol{w}_i, \boldsymbol{h}) - cos\angle(\boldsymbol{w}_i, \boldsymbol{w}_y))^2$$

The overall training loss:

$$\mathcal{L} = (1-\alpha)\mathcal{L}_{CE} + \alpha\mathcal{L}_{COS}$$

Experiment

Dataset	Height	Classes	Train	Val	Test
FGVC-Aircraft	3	100	3,334	3,333	3,333
CIFAR-100	5	100	45,000	5,000	10,000
iNaturalist2019	7	1010	187,385	40,121	40,737
tieredImageNet-H	12	608	425,600	15,200	15,200

Table 1. Statistics of the four datasets used in our experiments.

Type I: Backbone -> BN -> Linear(in_features, out_features) -> BN -> ELU

Type II

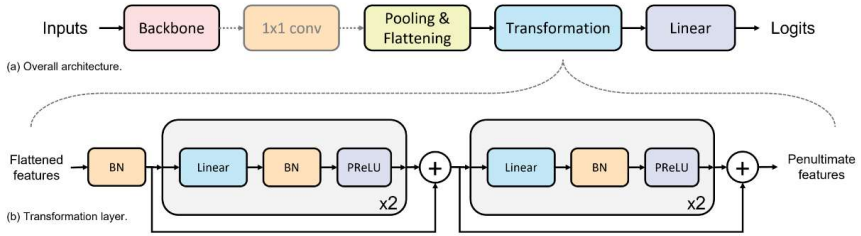


Figure 3. Illustration of the customized network architectures. (a) Top: overall network architecture of our approach, the 1x1 convolutional layer is only used in our type-II models. (b) Bottom: the proposed transformation layer, where BN is 1D batch norm layer.

Experiment

Evaluation Metrics

top-1 accuracy

average mistake severity

$$\mathbb{E}_{\mathbb{I}\{y_i != \hat{y}_i\}}[d_{y_i,\hat{y}_i}]$$

average hierarchical distance at k

$$\text{HierDist@}K = \mathbb{E}_{i \in N}[\frac{1}{K}\sum_{k=1}^{K} d_{y_i,\hat{y}_{i,k}}]$$

Experiment

Model	Method	Top-1 Accuracy ↑	Mistake Severity ↓	HierDist@1 ↓	HierDist@5 ↓	Hierdist@20 ↓
Type-I	cross-entropy	77.65 +/- 0.2635	2.34 +/- 0.0271	0.52 +/- 0.0102	2.25 +/- 0.0084	3.19 +/- 0.0045
	CRM [24]	77.63 +/- 0.2800	2.30 +/- 0.0255	0.51 +/- 0.0093	1.11 +/- 0.0077	2.18 +/- 0.0028
	Flamingo [7]	77.91 +/- 0.5733	2.31 +/- 0.0179	0.51 +/- 0.0137	2.07 +/- 0.0198	3.08 +/- 0.0094
	HAFeature [17]	77.49 +/- 0.3391	2.24 +/- 0.0158	0.51 +/- 0.0084	1.43 +/- 0.0108	2.64 +/- 0.0105
Type-II	cross-entropy	76.45 +/- 0.2207	2.43 +/- 0.0235	0.57 +/- 0.0106	2.35 +/- 0.0049	3.30 +/- 0.0030
	CRM [24]	76.48 +/- 0.2278	2.38 +/- 0.0175	0.56 +/- 0.0095	1.15 +/- 0.0074	2.20 +/- 0.0029
	Flamingo [7]	75.19 +/- 0.3188	2.31 +/- 0.0270	0.57 +/- 0.0043	2.42 +/- 0.0161	3.29 +/- 0.0105
	HAFeature [17]	76.44 +/- 0.1560	2.26 +/- 0.0290	0.53 +/- 0.0055	1.71 +/- 0.0130	2.84 +/- 0.0143
	HAFrame (ours)	77.71 +/- 0.2319	2.21 +/- 0.0108	0.49 +/- 0.0066	1.11 +/- 0.0018	2.18 +/- 0.0013

Table 3. Experiment results on CIFAR-100 dataset. The details of type-I and type-II models are included in the training config.

Model	Method	Top-1 Accuracy ↑	Mistake Severity ↓	HierDist@1 ↓	HierDist@5 ↓	Hierdist@20 ↓
Type-I	cross-entropy	70.68 +/- 0.2097	2.22 +/- 0.0103	0.65 +/- 0.0068	1.95 +/- 0.0043	3.37 +/- 0.0040
	CRM [24]	70.67 +/- 0.2095	2.16 +/- 0.0045	0.63 +/- 0.0057	1.17 +/- 0.0042	1.75 +/- 0.0033
	Flamingo [7]	70.11 +/- 0.1119	2.13 +/- 0.0063	0.64 +/- 0.0014	1.79 +/- 0.0126	3.28 +/- 0.0114
	HAFeature [17]	70.57 +/- 0.1645	2.13 +/- 0.0192	0.63 +/- 0.0045	1.55 +/- 0.2188	2.68 +/- 0.4208
Type-II	cross-entropy	70.44 +/- 0.1576	2.26 +/- 0.0071	0.67 +/- 0.0036	1.97 +/- 0.0060	3.40 +/- 0.0070
	CRM [24]	70.47 +/- 0.1363	2.21 +/- 0.0099	0.65 +/- 0.0029	1.18 +/- 0.0020	1.76 +/- 0.0016
	Flamingo [7]	70.13 +/- 0.1499	2.15 +/- 0.0061	0.64 +/- 0.0045	1.76 +/- 0.0037	3.31 +/- 0.0071
	HAFeature [17]	68.46 +/- 4.6278	2.21 +/- 0.1298	0.70 +/- 0.1501	1.50 +/- 0.1235	2.49 +/- 0.0842
	HAFrame (ours)	70.89 +/- 0.1213	2.04 +/- 0.0107	0.59 +/- 0.0033	1.14 +/- 0.0033	1.73 +/- 0.0023

Table 4. Experiment results on iNaturalist2019 dataset. The details of type-I and type-II models are included in the training config.

Experiment

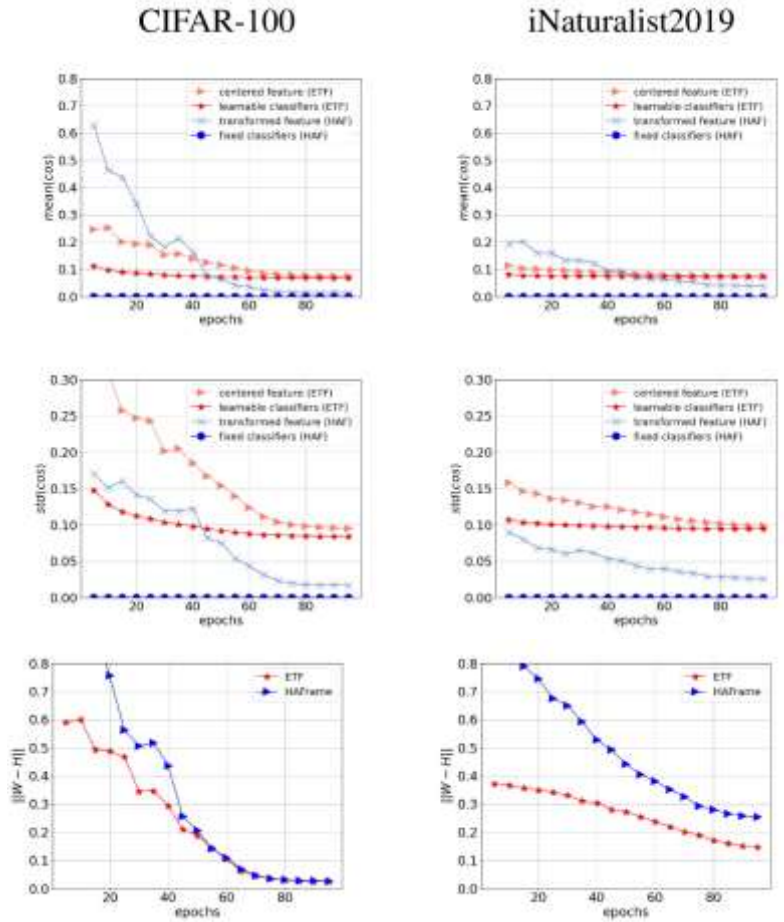
南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

dataset	model	$\mathcal{L}_{CE}$	$\mathcal{T}(.)$	HAF	$\mathcal{L}_{COS}$	Top-1 Acc ↑	Mistake Severity ↓	HieDist@1 ↓	HieDist@5 ↓	Hiedist@20 ↓
FGVC-Aircraft	ResNet50	✓	✗	✗	✗	79.18 +/- 0.5511	2.12 +/- 0.0240	0.44 +/- 0.0097	2.10 +/- 0.0033	2.67 +/- 0.0040
	ResNet50*	✓	✓	✗	✗	79.58 +/- 0.2727	2.15 +/- 0.0159	0.44 +/- 0.0067	2.11 +/- 0.0055	2.67 +/- 0.0034
	ResNet50*	✓	✓	✓	✗	79.18 +/- 0.5347	2.08 +/- 0.0299	0.43 +/- 0.0145	1.90 +/- 0.0056	2.55 +/- 0.0101
	ours	✓	✓	✓	✓	**80.49 +/- 0.4692**	**2.02 +/- 0.0381**	**0.39 +/- 0.0039**	**1.74 +/- 0.0027**	**2.45 +/- 0.0024**
CIFAR-100	WideResNet28	✓	✗	✗	✗	77.65 +/- 0.2635	2.34 +/- 0.0271	0.52 +/- 0.0102	2.25 +/- 0.0084	3.19 +/- 0.0045
	WideResNet28*	✓	✓	✗	✗	76.45 +/- 0.2207	2.43 +/- 0.0235	0.57 +/- 0.0106	2.35 +/- 0.0049	3.30 +/- 0.0030
	WideResNet28*	✓	✓	✓	✗	77.30 +/- 0.3798	2.37 +/- 0.0131	0.54 +/- 0.0111	1.59 +/- 0.0108	2.71 +/- 0.0147
	ours	✓	✓	✓	✓	**77.71 +/- 0.2319**	**2.21 +/- 0.0108**	**0.49 +/- 0.0066**	**1.11 +/- 0.0018**	**2.18 +/- 0.0013**
iNatura-list2019	ResNet50	✓	✗	✗	✗	70.68 +/- 0.2097	2.22 +/- 0.0103	0.65 +/- 0.0068	1.95 +/- 0.0043	3.37 +/- 0.0040
	ResNet50*	✓	✓	✗	✗	70.44 +/- 0.1576	2.26 +/- 0.0071	0.67 +/- 0.0036	1.97 +/- 0.0060	3.40 +/- 0.0070
	ResNet50*	✓	✓	✓	✗	**71.14 +/- 0.2245**	2.19 +/- 0.0132	0.63 +/- 0.0025	1.39 +/- 0.0021	2.20 +/- 0.0048
	ours	✓	✓	✓	✓	70.89 +/- 0.1213	**2.04 +/- 0.0107**	**0.59 +/- 0.0033**	**1.14 +/- 0.0033**	**1.73 +/- 0.0023**
tiered-ImageNet-H	ResNet50	✓	✗	✗	✗	73.63 +/- 0.1165	6.94 +/- 0.0208	1.83 +/- 0.0117	5.70 +/- 0.0192	7.34 +/- 0.0291
	ResNet50*	✓	✓	✗	✗	72.51 +/- 0.4317	6.95 +/- 0.0298	1.91 +/- 0.0338	5.69 +/- 0.0085	7.28 +/- 0.0082
	ResNet50*	✓	✓	✓	✗	73.54 +/- 0.2328	6.93 +/- 0.0274	1.83 +/- 0.0117	5.45 +/- 0.0026	6.82 +/- 0.0048
	ours	✓	✓	✓	✓	**74.00 +/- 0.3549**	**6.89 +/- 0.0251**	**1.79 +/- 0.0216**	**4.94 +/- 0.0118**	**6.15 +/- 0.0065**

Table 6. The ablation study results for FGVC-Aircraft (1st row), and CIFAR-100 (2nd row), iNaturalist2019 (3rd row), tieredImageNet-H (4th row). Each row in the table corresponds to the average results of 5 runs with a 95% confidence interval. Both ResNet50 and WideResNet28 are customized type-I models. The ResNet50*, WideResNet28*, and ours are customized type-II models.

adding the transformation layer alone does not necessarily improve the model performance. However, fixing the corresponding classifier weights to a HAFrame improves the average mistake severity of three datasets.

Experiment



Angular collapse

$$Avg_{1\le i<j\le K}(|cos\angle(\boldsymbol{x}_i,\boldsymbol{x}_j)-\hat{S}_{ij}|)$$
$$Std_{1\le i<j\le K}(|cos\angle(\boldsymbol{x}_i,\boldsymbol{x}_j)-\hat{S}_{ij}|)$$

Self-Duality

$$\left\|\frac{W}{\|W\|_F}-\frac{H}{\|H\|_F}\right\|_F$$

We observed that more training epochs (e.g., 200 or 350 epochs) lead to better self-duality on the larger datasets

Thank you