Supplementary Material: Adaptive Parametric Activation

Konstantinos Panagiotis Alexandridis[1], Jiankang Deng[1], Anh Nguyen[2], and Shan Luo[1,2]

[1] Huawei Noah's Ark Lab
{konstantinos.alexandridis,jiankang.deng}@huawei.com
[2] University of Liverpool, Liverpool L69 3BX, United Kingdom
{anguyen}@liverpool.ac.uk
[3] King's College London, London WC2R 2LS, United Kingdom
{shan.luo}@kcl.ac.uk

1 Representations' Quality

We evaluate the quality of the representations learned by the SE and APA* models using the recently proposed Neural Collapse framework [14].

Let $f_{k,j} \in \mathrm{R}^{\mathrm{d}}$ be the features of the penultimate layer, $k = \{1, 2, ..., K\}$ the class, n_k the number of samples in the class k and $n = \sum_{k=1}^{K} n_k$ the total number of samples in the dataset. Then the global feature f_G and class prototype $\bar{f}_k$ are:

$$f_G = \frac{1}{n} \sum_{k=1}^{K} \sum_{j=1}^{n_k} f_{k,j} \ , \bar{f}_k = \frac{1}{n_k} \sum_{j=1}^{n_k} f_{k,j} \tag{1}$$

The within-class covariance matrix $\Sigma_W \in \mathrm{R}^{\mathrm{d}\times\mathrm{d}}$ and between-class covariance matrix $\Sigma_B \in \mathrm{R}^{\mathrm{d}\times\mathrm{d}}$ are:

$$\begin{aligned} \Sigma_W :=& \frac{1}{n} \sum_{k=1}^{K} \sum_{j=1}^{n_k} (f_{k,j} - \bar{f}_k)(f_{k,j} - \bar{f}_k)^\top \\ \Sigma_b :=& \frac{1}{K} \sum_{k=1}^{K} (\bar{f}_k - f_G)(\bar{f}_k - f_G)^\top \end{aligned} \tag{2}$$

The Σ_W matrix shows how distant are individual features $f_{k,j}$ from their class prototype $\bar{f}_k$ and it is an indicator of feature compactness. The Σ_b matrix shows how distant are the class prototypes from the global feature, indicating the class separability. Using these matrices we measure the Neural collapse Variability $NC1$ according to [25] as follows:

$$NC1 := \frac{1}{K} trace(\Sigma_W \Sigma_b^\dagger) \tag{3}$$

where the † symbol denotes the pseudo inverse of Σ_b. $NC1$ measures the magnitude of the within-class covariance Σ_W compared to the magnitude of the between-class covariance Σ_b as explained in [25].

In practise, a low $NC1$ measure shows that the model has more compact features since $\Sigma_W \downarrow$ decreases and more separable class prototypes because the $\Sigma_b \uparrow$ increases. Having more compact features and more separable class prototypes make the representations better and enhance the classification results as shown empirically in previous works [18–20, 23].

Using Equation 3, we measure the $NC1$ of the deep features of the penultimate layer of SE and APA*, in Table 1 using ImageNet-LT test-set. As the results suggest, our APA* has lower $NC1$ measure for all backbones, showing that APA* produces superior representations that are more compact and seperable than the baseline. This provides another qualitative explanation why our APA* has better performance than SE.

Table 1: Neural Collapse NC1 measure, on ImageNet-LT test set. APA* has lower $NC1$ measure than the baseline, which indicates that it has learned superior representations.

Backbone	SE-$NC1 \downarrow$	APA*-$NC1 \downarrow$
ResNet-50	3.04	**2.71**
ResNeXt-50	3.38	**2.55**
ResNet-101	3.15	**2.69**
ResNet-152	3.24	**2.69**

2 Implementation Details

The implementation details of APA* and AGLU are shown in Table 2. For balanced ImageNet-1K, the λ parameters are initialised as random variables drawn from a Uniform distribution (U), with low parameter 0, and high parameter 1.0. The APA κ parameters are initialised with $U(0, 1)$ and the AGLU κ parameters are with initialised with $U(1, 1.3)$. For all other downstream tasks, that use a pretrained model, such as COCO, LVIS, Places-LT and V3Det, we don't re-initialise the κ and λ parameters and we simply load them from the pretained ImageNet1K model.

2.1 Stable APA implementation

During the development of APA, we found that it is more stable to use Softplus $s_f(z, \beta) = \frac{1}{\beta} \ln(1 + \exp(\beta z))$, than double exponents, when computing APA. Thus our stable code implementation is:

$$\eta_{ad}(z, \kappa, \lambda) = \exp(\frac{1}{\lambda} s_f(\kappa z - \ln(\lambda), -1)) \tag{4}$$

and it is equivalent to the APA used in the main paper.

Method	ImageNet-LT R50/X50	i-Naturalist18 R50	Places-LT R152	C100-LT R32	LVISv1 MRCNN-R50
Batch size	256	1024	256	512	16
Optimiser	SGD	SGD	SGD	SGD	SGD
LR	0.2	0.5	0.1	0.2	0.02
epochs	200	500	40	500	24
Weight Decay	1e-4	1e-4	5e-5	1e-3	1e-4
Norm Weight Decay	1e-4	0.0	5e-5	1e-3	1e-4
Bias Weight Decay	0.0	0.0	0.0	0.0	0.0
Attention Dropout	0.1	0.0	0.1	0.1	0.1
Mixup α	0.2	0.2	0.2	0.2	-
CutMix α	-	1.0	1.0	-	-
Label smoothing ϵ	-	0.1	0.1	-	-
Repeated Aug	-	✓	-	-	-
AutoAugment	✓	-	✓	✓	-
RandAugment	-	✓	-	-	-
Erasing prob	-	0.1	-	-	-
Cutout	-	-	-	✓	-
Cos. Cls. scale	16	16	learnable	learnable	N/A
Norm. Mask scale	N/A	N/A	N/A	N/A	learnable
Sampler	random	random	random	random	RFS
APA κ Init	U(-1,0)	U(0,1)	N/A	U(-1,0)	N/A
APA λ Init	U(0,1)	U(0,1)	N/A	U(0,1)	N/A
AGLU κ Init	U(1,1.3)	U(1,1.3)	N/A	U(1,1.3)	N/A
AGLU λ Init	U(0,1)	U(0,1)	N/A	U(0,1)	N/A

Table 2: Implementation details for Long-tailed Datasets, across various architectures.

2.2 AGLU derivatives

Proof of Eq. 9. Then the gradient of AGLU with respect to κ is:

$$\begin{aligned}\frac{\partial AGLU(x,\kappa,\lambda)}{\partial \kappa} &= \partial \frac{x \cdot (\lambda \exp(-\kappa x)+1)^{\frac{1}{-\lambda}}}{\partial \kappa} \\ &= x \frac{(\lambda \exp(-\kappa x)+1)^{(\frac{-1}{\lambda}-1)}}{-\lambda} \cdot (-\lambda x \exp(-\kappa x)) \\ &= x^2 \exp(-\kappa x) \frac{(\lambda \exp(-\kappa x)+1)^{(-\frac{-1}{\lambda})}}{\lambda \exp(-\kappa x)+1} \\ &= x^2 \frac{\eta_{\text{ad}}(x,\lambda,\kappa)}{\lambda + \exp(\kappa x)}\end{aligned} \tag{5}$$

Proof of Eq. 10. Then the gradient of AGLU with respect to λ is:

$$\begin{aligned}\frac{\partial AGLU(x,\kappa,\lambda)}{\partial \lambda} &= \partial \frac{x \cdot (\lambda \exp(-\kappa x)+1)^{\frac{1}{-\lambda}}}{\partial \lambda} \\ &= x \frac{(\lambda \exp(-\kappa x)+1)^{(\frac{-1}{\lambda}-1)}}{-\lambda} \cdot (\exp(-\kappa x)) \\ &= \frac{-x}{\lambda} \exp(-\kappa x) \frac{(\lambda \exp(-\kappa x)+1)^{(-\frac{1}{\lambda})}}{\lambda \exp(-\kappa x)+1} \\ &= \frac{-x}{\lambda} \frac{\eta_{\text{ad}}(x,\lambda,\kappa)}{\lambda+\exp(\kappa x)}\end{aligned} \tag{6}$$

Proof of Eq. 11. Then the gradient of AGLU with respect to λ is:

$$\begin{aligned}\frac{\partial AGLU(x,\kappa,\lambda)}{\partial x} &= \partial \frac{x \cdot (\lambda \exp(-\kappa x)+1)^{\frac{1}{-\lambda}}}{\partial x} \\ &= \eta_{\text{ad}}(x,\lambda,\kappa) + x \cdot \partial \frac{(\lambda \exp(-\kappa x)+1)^{\frac{1}{-\lambda}}}{\partial x} \\ &= \eta_{\text{ad}}(x,\lambda,\kappa) + x \frac{(\lambda \exp(-\kappa x)+1)^{(\frac{-1}{\lambda}-1)}}{-\lambda} \cdot (-\kappa\lambda \exp(-\kappa x)) \\ &= \eta_{\text{ad}}(x,\lambda,\kappa) + \kappa x \exp(-\kappa x) \frac{(\lambda \exp(-\kappa x)+1)^{(-\frac{1}{\lambda})}}{\lambda \exp(-\kappa x)+1} \\ &= \eta_{\text{ad}}(x,\lambda,\kappa) + \kappa x \frac{\eta_{\text{ad}}(x,\lambda,\kappa)}{\lambda+\exp(\kappa x)}\end{aligned} \tag{7}$$

3 Results

3.1 Experiments with AGLU and plain ResNets

In Table 3, we show the result of AGLU when it applied inside plain ResNet50 models, (i.e. without channel attention). As the Table suggests, by simply replancing the RELU with AGLU, our method consistently increases the performance of plain ResNet models.

Table 3: Top-1 accuracy on long-tailed classification datasets using ResNets.

Dataset	CIFAR100-LT		ImageNet-LT		iNaturalist
Imbalance factor	10	100	256		500
Model	ResNet-32		ResNet50	ResNeXt50	ResNet50
RELU	65.7	51.8	55.0	57.0	69.9
AGLU (ours)	**66.8**	**52.0**	**56.0**	**57.6**	**72.4**

3.2 Experiments with AGLU and Vision Transformers on ImageNet1K

We perform a preliminary experiment with Vision Transformer models such as ViT [6] and Swin [12] using ImageNet1K. We replace the GELU activation with AGLU in every feedforward layer and we keep all other settings the same. As shown in Table 4, AGLU performs comparably to GELU. We believe this is because the Self-Attention function makes the features smooth, by removing their harmonising components and it makes them more Gaussian-like [3,5,17,24]. Consequently, the Gaussial linear error unit, GELU, might be a good choice for the ViT network and our AGLU method has comparable performance.

Table 4: Results of AGLU using ViT models on ImageNet1K.

Model	Activation	epochs	top-1
ViT-B [6]	GELU	200	78.3
ViT-B [6]	AGLU	200	**78.5**
Swin-T [12]	GELU	100	78.7
Swin-T [12]	AGLU	100	**78.9**

3.3 Impact of Initialisation

In all of our experiments, we have initialised λ using the Uniform distribution with low parameter 0 and high parameter 1 as a default. Regarding the κ parameter inside AGLU, we initialise it to be close to 1.0, as this works best, as shown in Table 5. Regarding the κ parameter inside the attention layer, we found that initialising it with $Uniform(-1, 0)$ is slightly better than $Uniform(0, 1.0)$ as shown in Table 6.

Table 5: AGLU-κ parameter initialisation, using APA* ResNet50 backbone on ImageNet-LT. The λ is initialised with $Uniform(0, 1)$ by default.

AGLU - κ	top-1
$Uniform(0, 1)$	57.7
$Uniform(-2, 0)$	57.3
$Uniform(-3, 0)$	57.4
$Uniform(-2, 2)$	Failed
$Uniform(1, 1.3)$	**57.9**

Table 6: κ parameter initialisation inside the attention layer, using APA* ResNet50 backbone on ImageNet-LT. The λ is initialised with $Uniform(0, 1)$ by default.

APA - κ	top-1
$Uniform(0, 1)$	57.6
$Uniform(-1, 0)$	**57.9**

3.4 Channel specific λ and κ

We have also tried a variant that uses seperate λ and κ parameters for every channel. As shown in Table 7, this variant performs worse than using shared λ and κ parameters for the channels.

Table 7: Results with Channel Specific λ and κ, using APA* ResNet50 backbone on ImageNet-LT.

APA	top-1
Channel Specific	57.5
Channel Shared	**57.9**

3.5 Baseline enhancements

We show the detailed ablation study for ImageNet-LT in Table 8. First, the vanilla ResNet50 model trained for 100 epochs on ImageNet-LT achieves 44.4%. When we train for 200 epochs then it adds 1.5pp and switching from linear classifier to cosine classifier adds another 0.4pp. Stronger training techniques such Mixup [22], Auto-Augment [4] and weight decay tuning further boost the performance by 3.3pp. Post-calibrated Softmax [8] adds an additional 5.4pp and finally the Squeeze and Excite module [9] adds another 1.0pp reaching the final 56.0%. Most baseline performance comes from the PC-Softmax and the weight decay finetuning. On top of this strong baseline, our APA increases the performance by 1.0pp, showing its strong generalisability. Dropout and LayerNorm further increase the performance by 0.4pp and finally AGLU adds a respectable 0.5pp reaching 57.9% accuracy on ImageNet-LT. The absolute improvement of all modules is 13.5pp and our proposed methods, APA and AGLU, contribute by 1.5pp which is a relative 11% of the total absolute improvement.

3.6 Qualitative Results

In Figure 1, we show the learned parameters, when training with the balanced and imbalanced ImageNet. Regarding the λ inside the AGLU layers in (a), we see that both balanced-trained and imbalanced-trained networks prefer an all-pass filter for the early 2-3 layers, possibly, because the networks are uncertain

Table 8: Detailed Ablation Study, using ResNet50 on ImageNet-LT.

200 epochs	Cosine Classifier	SE-nets [9]	AutoAugment [4]	Mixup [22]	Weight Decay Tuning [1]	PCS [8]	APA (ours)	Dropout [7]	LayerNorm [2]	AGLU (ours)	ImagetNet-LT
											44.4
✓											45.9
✓	✓										46.3
✓	✓	✓									46.8
✓			✓								45.2
✓		✓	✓								45.9
✓	✓	✓	✓								46.6
✓	✓	✓	✓	✓							46.6
✓	✓		✓	✓	✓						49.6
✓	✓		✓	✓	✓	✓					55.0
✓	✓	✓	✓	✓	✓						51.7
✓	✓	✓	✓	✓	✓	✓					56.0
✓	✓	✓	✓	✓	✓	✓	✓				57.0
✓	✓	✓	✓	✓	✓	✓	✓	✓			57.3
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		57.4
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	**57.9**

which features to remove. Then in the intermediate layers, we observe smaller λ that corresponds to harder filters and in the final semantic layers we observe larger λ, possibly, because the network prefers smoother semantic features in order to have smoother classification boundaries. In (b), we see a 'down-down-up' κ-pattern in most bottlenecks, for both balanced and imbalanced ImageNet, showing that the networks prefer softer activations, at first, and harder activations before the residual connection. This indicates that the networks, first, keep most information inside the bottleneck's projections, and second, they disregard any redundant information, using harder activation, only before performing addition using the residual connection.

Finally, in the last bottlenecks, i.e. layers 45-50, the κ parameter diminishes, showing that the network prefers overly smooth activations, possibly, to enhance the classification using smoother classification boundaries.

Regarding the attention layers, the κ parameter in (d) dominates over the influence of λ in (c), showing that hard channel attention is more preferable than soft channel attention for all layers.

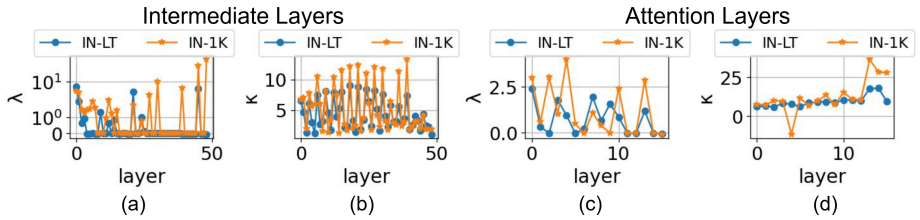

Fig. 1: Visualisations of the learned λ and κ parameters for balanced ImageNet1K (IN-IK) training in blue, and imbalanced ImageNet-LT (IN-LT) training in blue.

Visualisations on Imagenet-LT We further show more qualitative results on ImageNet-LT with ResNet50 backbone. On the left subfigure, we show the model's highest prediction marked with F,C,R that stands for frequent, common and rare class respectively and the Grad-cam activation [16]. On the right subfigure, we show the last layer's channel attention signal and its corresponding entropy denoted with (E). As the Figure shows, APA* produces higher entropy attention signals than the baseline and predicts both frequent and rare classes correctly.

3.7 Calibration results

Calibration is an important property of models, since it reassures that the confidence of the prediction matches the actual accuracy. When models are not calibrated, then they give wrong predictions with high confidence score (over-confident models) or make correct predictions with low confidence score (under-confident models). In both situations, the miscalibrated models cannot help in the decision making process because their predictions do not reflect their actual accuracy.

In practice in long-tailed learning, the use of complex augmentations and regularisations like mixup, cutmix, label-smoothing, auto-augment and cosine classifier may improve the accuracy but it also reduces the confidence of the model due to over regularisation. As shown in Figure 3 (left-subfigure), SE-Resnet50 is under-confident due to the usage of complex training that includes heavy augmentations and regularisations. When APA* is applied, it reduces the Expected Calibration Error (ECE) as shown in Figure 3 (right-subfigure) for all backbones.

3.8 Next textual token prediction experiment

We perform one preliminary next-token prediction experiment using GPT2 [15] and the FineWeb-Edu [13] subset that contains 10 billion GPT2 tokens. The model is based on the GPT2 smallest architecture, which contains 117M parameters, and the code implementation follows [10]. We train the GPT2 model for one epoch, using 8 V100 GPUs, a total batch size of 0.5M tokens, learning

Fig. 2: Comparative Results between the SE-ResNet50 (baseline) and APA*-ResNet50 (ours) with respect to the activations (left) and the attention entropy (right). F,C,R denote frequent, common and rare samples from ImageNet-LT. Our method produces attention signals that have significantly larger entropy than the baseline for both frequent and rare classes.

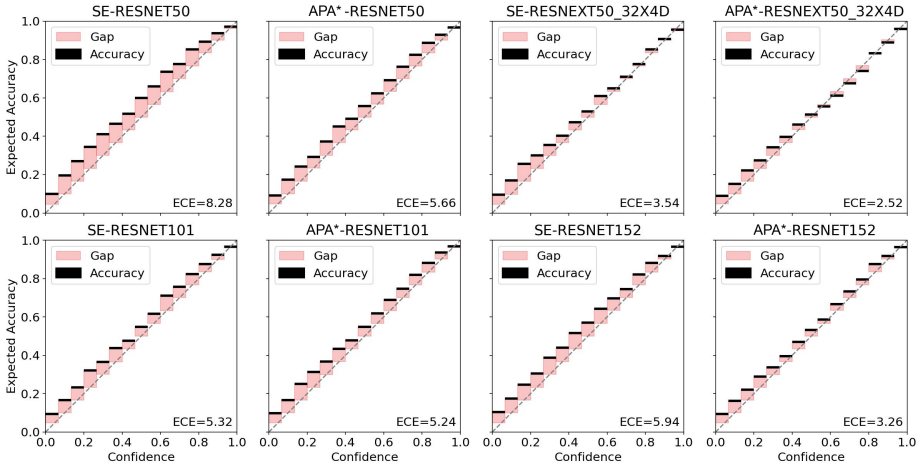

Fig. 3: Calibration results using ResNets on ImageNet-LT. SE (left) is underconfident, i.e., its confidence scores are lower than its actually accuracy due to over-regularisation. Our APA* (right) reduces the ECE and makes more accurate predictions with higher confidence than SE.

rate $6e-4$ and Adam optimizer [11] with momentum. We test the model on the HellaSwag benchmark [21] using zero-shot evaluation. To apply AGLU with GPT2, we simply switch the GELU activation with AGLU inside all MLP layers of the transformer. As the results suggest in Table 9, our AGLU increases the performance of GPT2 by 0.4%, showing that AGLU could be a good alternative for text-classification.

Table 9: Comparative results using GPT2 smallest model and HellaSwag benchmark.

Method	Accuracy
GELU	31.0
AGLU	**31.4**

References

1. Alshammari, S., Wang, Y.X., Ramanan, D., Kong, S.: Long-tailed recognition via weight balancing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6897–6907 (2022)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Bai, J., Yuan, L., Xia, S.T., Yan, S., Li, Z., Liu, W.: Improving vision transformers by revisiting high-frequency components. In: Computer Vision–ECCV 2022: 17th

European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV. pp. 1–18. Springer (2022)
4. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 113–123 (2019)
5. Dong, Y., Cordonnier, J.B., Loukas, A.: Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In: International Conference on Machine Learning. pp. 2793–2803. PMLR (2021)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=YicbFdNTTy
7. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
8. Hong, Y., Han, S., Choi, K., Seo, S., Kim, B., Chang, B.: Disentangling label distribution for long-tailed visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6626–6636 (2021)
9. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
10. Karpathy, A.: build-nanogpt. https://github.com/karpathy/build-nanogpt (2024)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
13. Lozhkov, A., Ben Allal, L., von Werra, L., Wolf, T.: Fineweb-edu (May 2024). https://doi.org/10.57967/hf/2497, https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu
14. Papyan, V., Han, X., Donoho, D.L.: Prevalence of neural collapse during the terminal phase of deep learning training. Proceedings of the National Academy of Sciences **117**(40), 24652–24663 (2020)
15. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
16. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
17. Wang, P., Zheng, W., Chen, T., Wang, Z.: Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=O476oWmiNNp
18. Xie, L., Yang, Y., Cai, D., He, X.: Neural collapse inspired attraction–repulsion-balanced loss for imbalanced learning. Neurocomputing **527**, 60–70 (2023)
19. Yang, Y., Chen, S., Li, X., Xie, L., Lin, Z., Tao, D.: 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)

20. Yang, Y., Yuan, H., Li, X., Lin, Z., Torr, P., Tao, D.: Neural collapse inspired feature-classifier alignment for few-shot class incremental learning. arXiv preprint arXiv:2302.03004 (2023)
21. Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., Choi, Y.: Hellaswag: Can a machine really finish your sentence? arXiv preprint arXiv:1905.07830 (2019)
22. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
23. Zhong, Z., Cui, J., Yang, Y., Wu, X., Qi, X., Zhang, X., Jia, J.: Understanding imbalanced semantic segmentation through neural collapse. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19550–19560 (June 2023)
24. Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., Hou, Q., Feng, J.: Deepvit: Towards deeper vision transformer. arXiv preprint arXiv:2103.11886 (2021)
25. Zhu, Z., Ding, T., Zhou, J., Li, X., You, C., Sulam, J., Qu, Q.: A geometric analysis of neural collapse with unconstrained features. Advances in Neural Information Processing Systems **34**, 29820–29834 (2021)