Gramian Attention Heads are Strong yet Efficient Vision Learners – Appendix –

Network	CUB-200	Food	Cars	Aircraft	Flowers
R50	79.2±1.7	86.9±0.2	90.4±1.0	86.2±2.0	99.0±0.4
GA-R50	**81.9±0.6**	**87.7±0.4**	**91.3±1.1**	86.6±1.4	99.2±0.2
GA-R50 (w/o heads)	81.4±1.0	87.1±0.1	90.7±1.0	**86.8±1.4**	**99.3±0.2**

Table A: **Transfer learning results on FGVC datasets**. We compare the accuracy of the baseline ResNet50 and GA-ResNet50 with or without heads. We report the averaged accuracies with the standard deviation to show the accuracy robustness across diverse hyper-parameter settings for each dataset. All the accuracies are reported by training and evaluation with 224×224 images. We observe that our approach demonstrates enhanced transferability. Surprisingly, even our model without heads (*i.e.*, the backbone itself) exhibits improved transferability as well.

Dataset	Accuracy gains (%p)
CIFAR (depth=29/65/110)	+0.49 / +0.85 / +0.24
ImageNet / CUB / Food	+0.28 / +2.2 / +0.2
Car / Aircraft / Flower	+0.3 / +0.6 / +0.3

Table B: **Impact of our models without heads.** We study the backbone performance after eliminating heads. The numbers indicate top-1 accuracy gains over each baseline, which is trained with the identical setting to ours. This reveals our proposed method consistently improves the backbone's expressiveness across different datasets, detaching heads after training.

A. Experiment (cont'd)

A.1. Fine-grained Visual Classification

Training setup. To further investigate our pretrained models' transferability, we finetune the ImageNet-pretrained GA-ResNet50 on the fine-grained visual classification (FGVC) datasets. We employ five datasets, including CUB-200 [13], Food-101 [1], Stanford Cars [7], FGVC Aircraft [10], and Oxford Flowers-102 [11]. We grid-search the hyper-parameters similarly to [6, 5] and follow the provided training regime for finetuning. We use the SGD optimizer with 20k iterations train networks and 224×224 center-cropped images from the downsized one to 256 on its shorter side. We also finetune the ImageNet-pretrained ResNet50 on each dataset as baselines using identical grid-searches to exhibit maximal performance. Note that we report the accuracy at the final epochs rather than picking up the peak accuracy. We do not compare with the finetuning performance of other backbones due to the inherent differences in model size and finetuning training setup. As shown in Table A, each of GA-ResNet50s consistently outperforms their respective baseline counterparts.

Finetuining models without heads. We conjecture that our models would have empowered backbones (*i.e.*, the models without head classifiers) having improved transferability. To validate this, we report the transfer learning performance of the finetuned backbone, which is identical to ResNet50 without the heads. Table A shows that GA-ResNet50 (without head classifiers) enjoys consistent extra accuracy gains in Table A. We presume that our proposed method encourages the early layers (*i.e.*, input-side layers) to learn more transferable representations due to the proposed lightweight heads that possess a few trainable parameters. We believe this shows a potential of utilizing our GA-networks as a partial network without using heads at inference for further efficiency. We will give more results about employing partial networks in the later section.

B. Additional Experimental Studies

We conduct additional empirical studies with our proposed method. First, we showcase the capability of backbones that have no heads and models with only a single head by randomly removing all other heads. Second, we present comparative experiments with an existing multi-head neural network [8].

B.1. Deploying Partial Networks

Our proposed method enables the deployment of partial networks from the overall learned network, enhancing efficiency (*i.e.*, using the backbone alone or the network with fewer heads). In conjunction with Table A, the CIFAR and ImageNet results in Table B offer additional evidence that our backbones experience substantial improvement without heads, all without incurring extra computational demands. As aforementioned, using lightweight heads contribute to this improvement. We further speculate that this outcome arises due to the augmented gradients originating from multiple heads, which are learned through the proposed method. Furthermore, we argue that our decorrelation loss augments

Network	Depth	FLOPs (G)	#Params (M)	Top-1 err (%)	Top-5 err (%)
R50	29	0.05	0.34	26.1	6.5
	65	0.10	0.71	22.0	4.9
	110	0.17	1.17	19.8	4.4
GA-R50	29	0.05	0.36	**24.8**	**6.3**
	65	0.11	0.76	**21.2**	**4.8**
	110	0.18	1.26	**19.6**	**4.4**

Table C: **Impact of our models using only a single head.** We report a performance comparison of our models with a single head classifier with the ResNet baselines. The results show that only a single head classifier with negligible extra computational costs gives consistent and significant performance improvements.

the gradients again, promoting less-correlatedheads.

We adjust our models using only a single head classifier upon the baseline. We remove all the other heads but remaining a single head that is randomly chosen. The single head at the top of a backboneincurs minimal computational costs compared to the backbone itself yet achieves significant performance improvement, as shown in Table C. In practice, the number of head classifiers can be adjusted to balance the accuracy, memory, and latency under resource limits.

B.2. Comparison with Multiple Feature Learning

Finally, we conduct additional experiments comparing with a prior multiple-feature learning method, which learns multiple features and aggregates. This is to show whether our method with lightweight heads actually works better than the method with heavy and complicated heads. Since such architectures [9, 3] were aimed at different tasks, we choose a milestone work [8] that also trains multiple high-level features from multiple branches for comparison.

The branches in ONE-E [8] appear similar to our head classifiers; however, ONE-E uses a copy of fractions in its backbone, resulting in overall heavy computational costs. Moreover, those branches are positioned differently compared with ours. ONE-E training highly relies on knowledge distillation to learn similar features among the branches, where the concept is completely distinct from ours. We argue that the reported improvements in the paper may stem from the heavy branches; they could learn expressive representations but are highly correlated to each other.

To ensure a fair comparison, we employ the identical architecture proposed in the paper [8] for training, which is found in the publicly released codebase[1], where there are three branches from the middle layer of ResNets. ResNet32 (R32) and ResNet110 (R110) are used for experiments, the standard network architectures for CIFAR training [4, 14]. We train the models for ONE-E and ours with identical

[1] https://github.com/lan1991xu/one_neurips2018

Method	FLOPs (G)	#Params (M)	Top-1 err (%)	Top-5 err (%)
ONE-E + R32	0.12	1.19	24.0	5.6
GA -R32	**0.08**	**0.75**	**21.9**	**5.1**
ONE-E + R110	0.29	2.96	19.9	4.3
GA -R110	**0.22**	**2.04**	**19.0**	**3.9**

Table D: **Comparison with the multiple feature learning method.** We perform an experimental comparison of our method with ONE-E [8]. Two baselines ResNet32 (R32) and ResNet110 (R110) are used, and ours consistently outperform the counterparts.

Network	#Params (M)	Memory		
		128	256	512
ViT-S	25.6	157.6	231.1	378.1
GA-ViT-S	41.3	179.2	252.7	399.7

Table E: **Memory usage by batch size (*i.e.* 128, 256, and 512).** We measure the memory usage of the input image tensor and parameters for ViT [2, 12] models.

training setups. Table D shows that our models with the same number of head classifiers achieve better performance with extremely less computational demands.

B.3. Memory usage

We measure the additional memory usage by our proposed method. As shown in Table E, the parameter overhead of our method is not severe, so the additional memory usage of them is manageable. Since this memory usage is mostly proportional to the parameters, other models with GA will show similar trends.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 1

[2] 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. In *ICLR*, 2020. 2

[3] Xianzhi Du, Tsung-Yi Lin, Pengchong Jin, Golnaz Ghiasi, Mingxing Tan, Yin Cui, Quoc V Le, and Xiaodan Song. Spinenet: Learning scale-permuted backbone for recognition and localization. In *CVPR*, 2020. 2

[4] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *CVPR*, 2017. 2

[5] Dongyoon Han, Sangdoo Yun, Byeongho Heo, and YoungJoon Yoo. Rethinking channel dimensions for efficient model design. In *CVPR*, 2021. 1

[6] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *CVPR*, pages 2661–2671, 2019. 1

[7] Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. *In Second Workshop on Fine-Grained Visual Categorization*, 2013. 1

[8] Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. In *NeurIPS*, 2018. 1, 2

[9] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *ICCV*, 2017. 2

[10] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv*, 2013. 1

[11] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 1

[12] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 2021. 2

[13] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1

[14] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 2