Meta-attention for ViT-backed Continual Learning
–*Supplementary Material*–

Mengqi Xue[1], Haofei Zhang[1], Jie Song[1,†], Mingli Song[1,2]
[1]Zhejiang University
[2]Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, Zhejiang University
{mqxue,haofeizhang,sjie,brooksong}@zju.edu.cn

1. Implementation Details

We give more implementation details of our experiments in this section.

1.1. Datasets

We give the summary statistics of all involved datasets in Table 1, including six small datasets: CUB [16], Stanford Cars [7], FGVC-Aircraft [10], CIFAR-100 [8], Sketches [4] and WikiArt [13], and two large-scale datasets: ImageNet [3] and Plcaces365 [20].

Task	Train	Validation	Classes
CUB	5994	5794	200
Cars	8144	8041	196
FGVC	6667	3333	100
WikiArt	42129	10628	195
Sketches	16000	4000	250
CIFAR-100	50000	10000	100
Places365	1803460	36500	365
ImageNet	1281144	50000	1000

Table 1. Detailed statistics of adopted eight datasets.

1.2. Training Settings

We follow most of the training strategies used in DeiT [15] and implement the proposed method based on its official code [1] with PyTorch [12] on two Nvidia Tesla A100 GPUs.

For data augmentation, extensive tricks are put to use, such as mixup [19], cutmix [18], Rand-Augment [2], repeated augmentation [1, 5], and random cropping. All input images are resized to 224×224 pixels to ensure consistency with images of ImageNet. The official pre-trained weights of the ImageNet task are adopted as the well-initialization weights of the old task.

1	CUB	Cars	FGVC	WikiArt	Sketches	CIFAR
2	Cars	CIFAR	CUB	Sketches	WikiArt	FGVC
3	FGVC	Cars	Sketches	CIFAR	WikiArt	CUB
4	WikiArt	CIFAR	Sketches	CUB	FGVC	Cars
5	Sketches	CUB	FGVC	Cars	CIFAR	WikiArt
6	FGVC	Cars	WikiArt	CIFAR	CUB	Sketches

Table 2. Six task sequences. "CIFAR" represents CIFAR-100 dataset for simplicity.

For training, all models are trained for 30 epochs (5 warm-up epochs) on two GPUs with a batch size of 256. AdamW [9] is employed as the optimizer using cosine linear-rate scheduler with a weight decay of $5e^{-2}$. The initial learning rates of backbones (including new classifiers and added linear layers in Adaptor-Bert [6],) and MEAT masks are $\frac{batchsize}{1024} \times 5e^{-4}$ and $\frac{batchsize}{1024} \times 0.1$. The learning rate of the masks introduced Piggyback [11] is $\frac{batchsize}{1024} \times 5e^{-4}$, following the same learning rate used in Piggyback. Mean accuracy is taken over six random task orders using five seeds, which are 226, 580, 1028, 2685, and 3486 respectively. The detail task sequences is listed in Table 2.

1.3. Baseline Parameters

Table 1 in the main text presents our main implementation results, and we give the descriptions of hyperparameters in baselines. Classifier and Finetuning baselines don't introduce any extra parameters. The former trains new classifiers only, and the latter retrains whole models. For other methods, the hyperparameters in both our method and competitors are set by grid search. For LwF baseline, the coefficient multiplied with the classification loss of the new task is 1; the coefficient multiplied with each distillation loss is 2. For Piggyback [11], the real-value mask is initialized with value 0.01, and the default threshold of the binazier is $5e^{-3}$. Specially, for HAT [14], the gated task embeddings are multiplied with token embeddings after the FFN block at each

†Corresponding author
[1]https://github.com/facebookresearch/deit

New Task	ImageNet			Places365			Random		
	DeiT-Ti	DeiT-S	T2T-ViT-12	DeiT-Ti	DeiT-S	T2T-ViT-12	DeiT-Ti	DeiT-S	T2T-ViT-12
CUB	71.16	81.53	69.90	65.37	71.33	50.99	46.05	49.10	26.15
Cars	53.42	77.20	61.90	48.64	63.78	53.98	16.27	18.29	11.52
FGVC	52.69	65.69	53.55	46.79	60.43	44.23	14.35	15.51	12.46
WikiArt	64.63	73.43	61.20	60.7	69.42	58.64	43.85	35.57	32.54
Sketches	70.73	76.68	74.75	60.11	67.99	68.64	39.80	18.79	45.24
CIFAR-100	78.13	85.93	77.42	73.19	80.03	71.76	66.05	72.71	33.10

Table 3. Accuracy (%) on new tasks added on different old tasks as initializations. “ImageNet” and “CUB” are well-trained weights on the ImageNet dataset and the CUB dataset tasks served as initializations, separately. And “Random” denotes that three vision transformers are randomly initialized.

encoder layer. It is important to note that HAT has to store task-specific embeddings. For the initial ImageNet task, we don’t train ViTs on ImageNet and directly utilize the official open-source pre-trained weights. As a result, the experiments of HAT lack the embeddings of ImageNet. Given this circumstance, the performance on ImageNet of HAT has been omitted. In Table 1 in the main text, we use “N/A” to denote that the performance on the ImageNet dataset of the HAT baseline is omitted.

2. Additional Abaltion Experiments

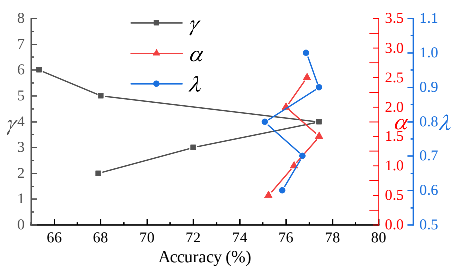

Figure 1. Sensitivity analysis on CIFAR-100 with ViT-Ti.

Sensitivity analysis of hyperparameters in MEAT. We use grid search for tuning all hyperparameters. Here we show the results of used hyperparameter optimization with grid search in Fig. 1. It can be seen that the results of MEAT are less sensitive to α and λ and are more vulnerable to γ.

	Individual	Classifier	LwF	Piggyback	HAT	BERT-adaptor	MEAT
Std	1.37	1.28	1.47	1.29	1.26	1.31	1.23

Table 4. Standard deviation on CIFAR-100 with ViT-Ti.

Uncertainty. Standard deviation is provided as an indicator of uncertainty averaged over all runs in Table 4. It can be concluded the proposed MEAT shows low standard deviation on the CIFAR-100 dataset compared to other baseline methods.

Model	#	CUB	CIFAR100	γ	CUB	CIFAR100
DeiT-Ti	3	68.31	76.99	2	70.33	77.89
	5	69.90	77.16	4	71.16	78.13
	12	71.16	78.13	6	68.39	75.34
DeiT-S	3	80.94	84.77	2	79.54	83.46
	5	81.29	84.80	4	81.53	85.93
	12	81.53	85.93	6	80.21	83.90

Table 5. Accuracy (%) of different numbers (#) of transformer layers applied with the MEAT masks and different initial values γ of t^i on the CUB dataset and the CIFAR-100 dataset.

Different Initialization. In our main experiments, we adopt a well-initialization transformer-based model to provide the knowledge of the old task, specifically the ImageNet task. In this subsection, we want to investigate the influence of different initialization via the weights of different old tasks. As shown in Table 3, ImageNet, Places365, and Random are three types of model initializations for DeiT-Ti [15], DeiT-S [15], and T2T-ViT-12 [17]. In other words, they are three old initial tasks. It can be observed that ImageNet enjoys the most superior performance on new tasks compared to the other two old tasks. And Places365 performs worse than the ImageNet initialization, as the images in the Places365 show large domain shifts from the target small tasks. Moreover, both ImageNet and Places365 achieve much better performance than Random. These observations verify that a good initialization can boost the performance of MEAT by a large margin cause its weights are trained on diverse image data like ImageNet and Places365. Meanwhile, the old task with a closer domain to target tasks, like ImageNet, tends to serve a good initial task. Results on Random indicate that appropriate pro initialization is significant for vision transformers in MEAT.

Influence of Hyperparameters. In this subsection, we provide a qualitative analysis of different choices on hyperparameters used in our method. Table 5 first analyses performance when adding the MEAT masks on tokens and neurons in different encoder layers on the CUB dataset and the CIFAR-100 dataset, and we find that the more masks are inserted, the better the results. When all encoder layers are added with the attention masks (12 layers), the model shows the best performance of learning two new tasks. The influence of different initial values γ, which is the initial weights of t^i of the masks, is also investigated. As shown in Table 5, it can be observed in Table 5 that both too-large and too-small initial values lead to performance deterioration. Consequently, a medium value is appropriate for our proposed MEAT masks. For clarity, the specific locations (transformer layers) added with MEAT masks in Table 5 are listed as: (a) 3: layer-4, layer-7, layer-10; (b) 5: layer-2, layer-4, layer-6, layer-8, layer-10; (c) 12: layer-1, layer-2, layer-3, layer-4, layer-5, layer-6, layer-7, layer-8, layer-9, layer-10, layer-11, layer-12.

3. Additional Analysis and Discussion

We visualize the trained binary masks on tokens of each encoder layer in DeiT-Ti, DeiT-S, T2T-ViT-12, as shown in Figure 2, Figure 3, and Figure 4, separately. In each figure, the example images in each row are from CUB, Car, FGVC, WikiArt, Sketches, CIFAR-100. The results keep the same as Section 4.4 in the main text. It can be observed that all three transformers tend to isolate many image tokens at the first layer. Then at the shallow layers they activate most of the image tokens at the shallow layers. With the deepening of layers, more tokens are isolated and the models put more attention on the regions where the targets are more likely to appear, for example, the central patches are activated with a higher probability at deep layers. It also can be verified that the bigger model, DeiT-S has a tendency to activate more tokens at shallow layers than the other two smaller models.

References

[1] Maxim Berman, Hervé Jégou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. Multigrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019. 1

[2] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, pages 702–703, 2020. 1

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 1

[4] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *TOG*, 31(4):1–10, 2012. 1

[5] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *CVPR*, pages 8129–8138, 2020. 1

[6] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, pages 2790–2799. PMLR, 2019. 1

[7] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, pages 554–561, 2013. 1

[8] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[10] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1

[11] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, pages 67–82, 2018. 1

[12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32:8026–8037, 2019. 1

[13] Babak Saleh and Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *arXiv preprint arXiv:1505.00855*, 2015. 1

[14] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, pages 4548–4557. PMLR, 2018. 1

[15] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 1, 2

[16] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 1

[17] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. 2

[18] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 1

[19] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1

[20] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *PAMI*, 40(6):1452–1464, 2017. 1

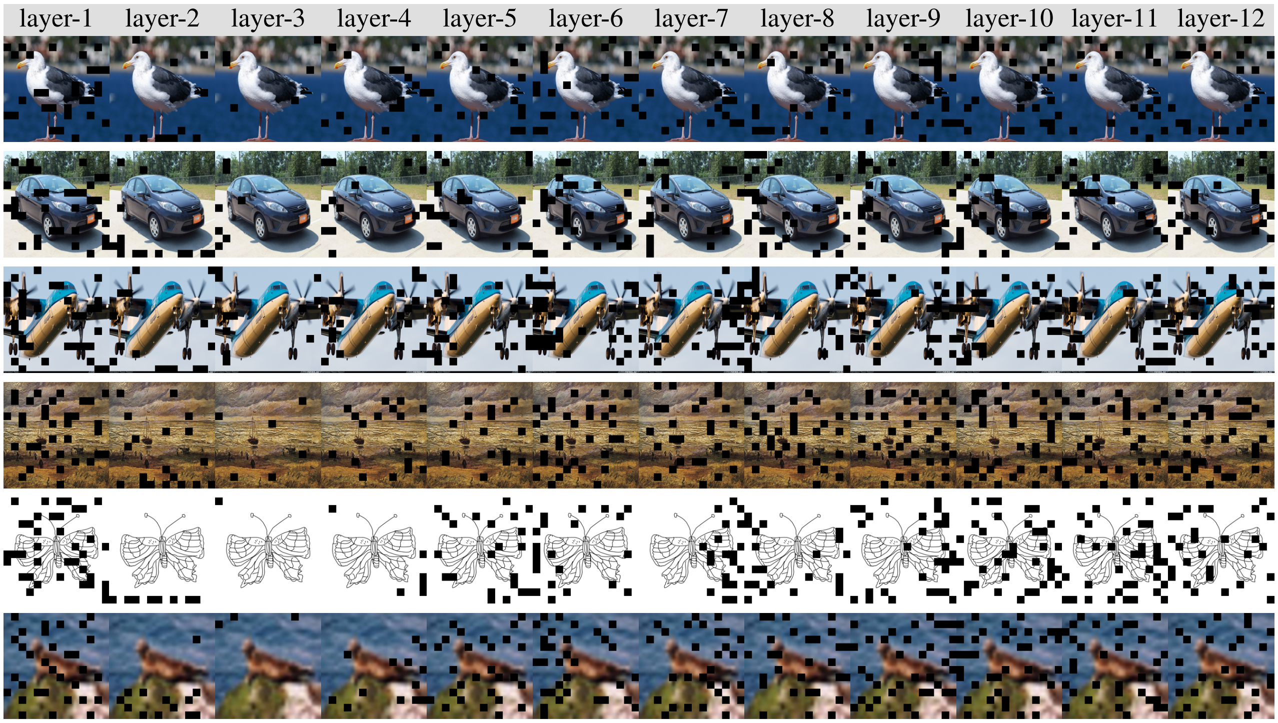

Figure 2. Visualization of the trained MEAT masks (on neurons) on the example images of six datasets at each encoder layer in DeiT-Ti.

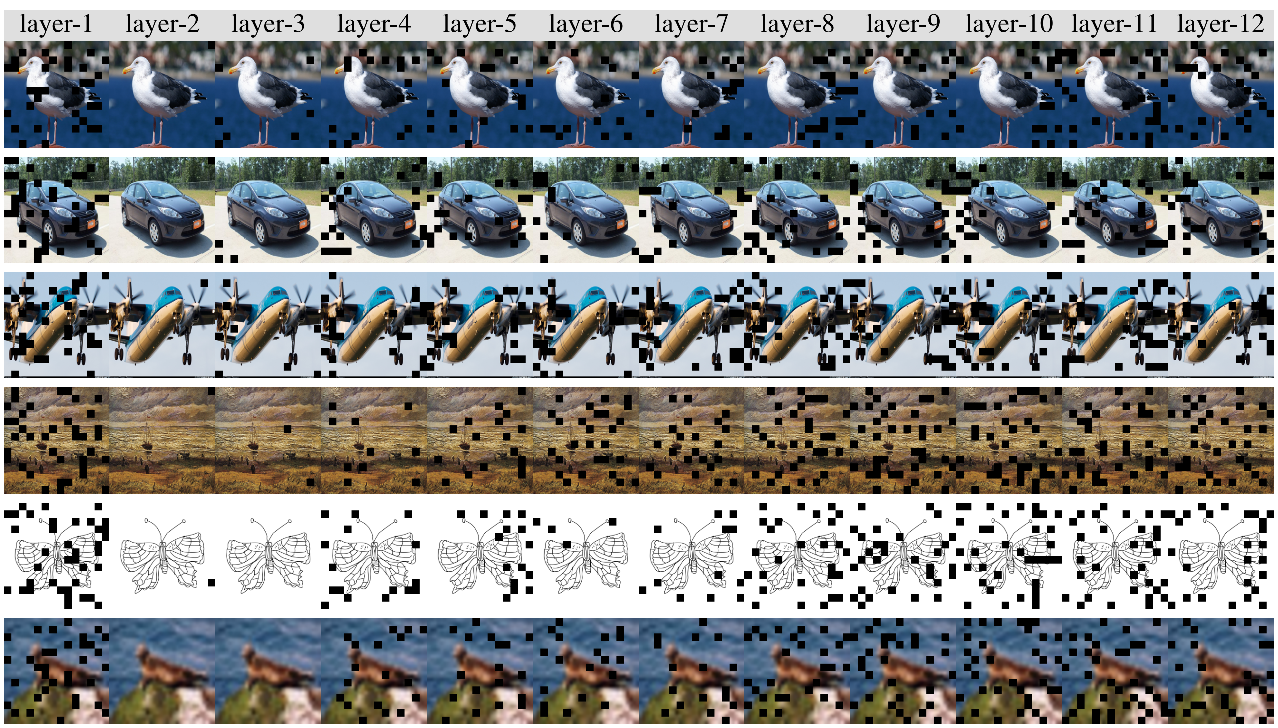

Figure 3. Visualization of the trained MEAT masks (on neurons) on the example images of six datasets at each encoder layer in DeiT-S.

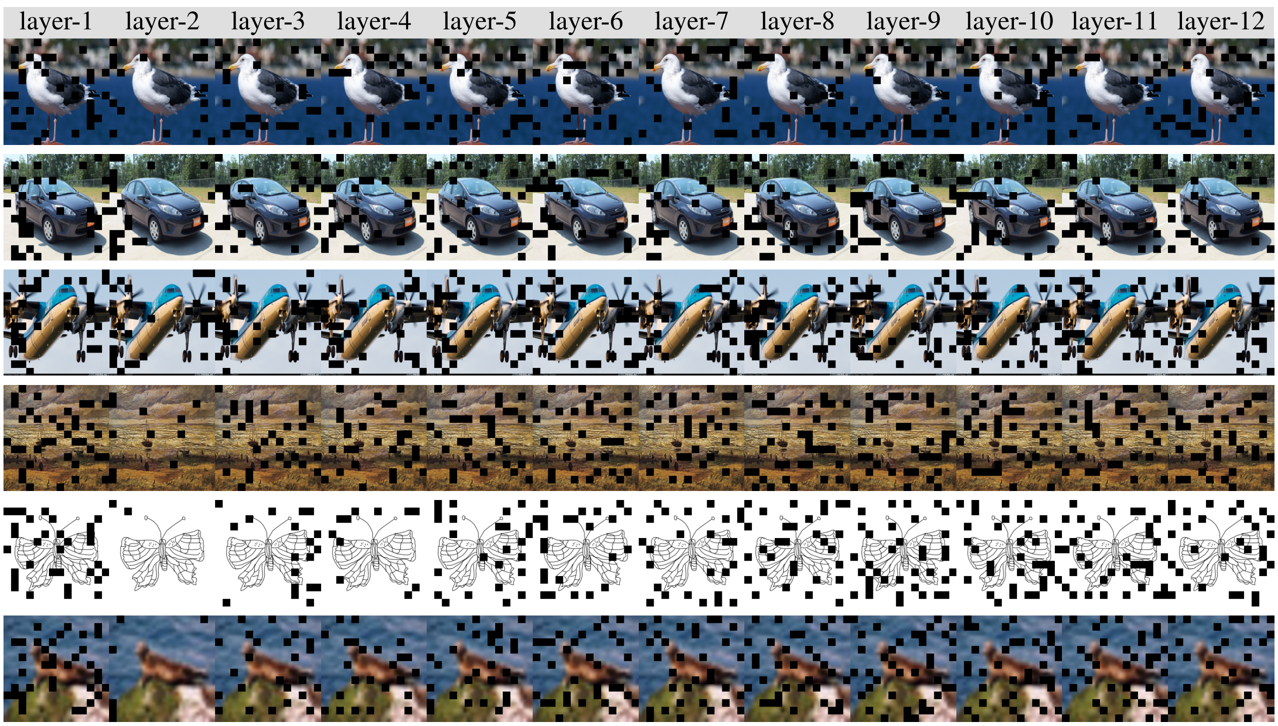

Figure 4. Visualization of the trained MEAT masks (on neurons) on the example images of six datasets at each encoder layer in T2T-ViT-12.