Meta-attention for ViT-backed Continual Learning

1、Motivation

过往的用于CNN的增量学习方法在ViT中并不够适用。

作者设计了一种基于mask method的ViT IL方法。之所以选择mask method，是因为：

1）mask方法为每个task选用特定的参数，因此非常适配IL

2）mask方法对task对顺序并不敏感

3）mask方法避免了昂贵对数据存储开销

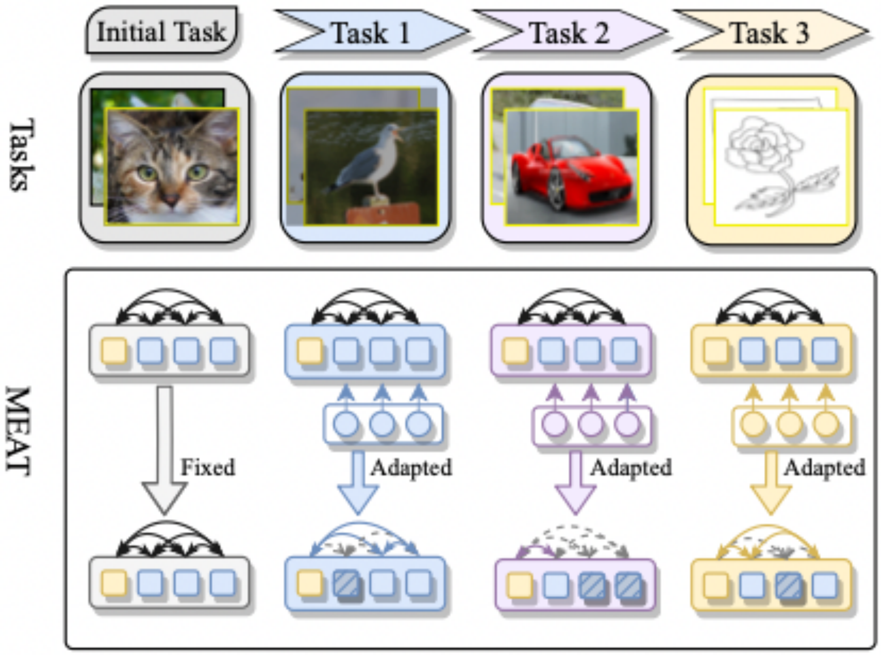


Figure 1. The proposed MEAT for task continual learning in the MHSA block with vision transformers. With the increase of new tasks, MEAT dynamically assigns attention masks to generate task-specific self-attention patterns per task.

基于此，作者设计了Meta-Attention方法。

2、Approach

The proposed MEta-ATtention (MEAT) aims to dynamically adapt the standard token interaction pattern to the new tasks via putting attention to self-attention.

2.1 Attention to Self-attention

原本一个head的self-attention计算方法为：

$$\text{head}_h = \mathbf{\Psi}_h \mathbf{V}_h = \sigma\left(\mathbf{A}_h\right) \mathbf{V}_h = \sigma\left(\frac{\mathbf{Q}_h \mathbf{K}_h^\top}{\sqrt{d_k}}\right) \mathbf{V}_h,$$

作者在这里引入了一个mask，在计算的时候掩盖一部分的token，attention的计算方法表示为：

$$\Psi_h^i=\left[\Psi_h^{i,j}\right]_{j=1}^n=\left[\frac{m^j\exp\left(A_h^{i,j}\right)}{\sum_{s=1}^n m^s\exp\left(A_h^{i,s}\right)}\right]_{j=1}^n .$$

这个过程可以用下图表示。标准的Token interaction中token的信息交互是不受限制的。但adaptied token interaction中，通过mask隐去了部分token之间的交互。

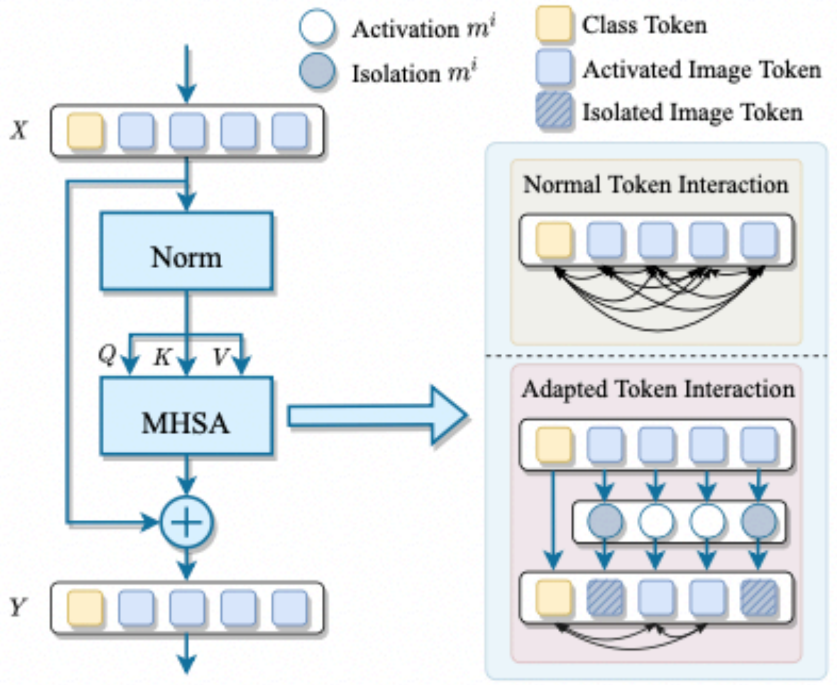


Figure 2. Illustration of the working mechanism of MEAT in the MHSA block of a transformer encoder layer. In the standard token integration, all the tokens interact with each other without limit. MEAT proposes attention masks to modify this communication pattern by dynamically activating and isolating image tokens. X and Y denote the input and the output sequence. Q, K and, V represent query, key, and value for the MHSA module.

看起来，每个mask是针对task来设计的。但是读到这里有问题：**掩盖一些token 不会掉点吗？以及为什么掩盖token这个方法可行呢？**

在实现上， a binary value MEAT mask is adopted to replace the former continuous value mask. For the token i, a binary variable, the attention entry $m_i \in \{0, 1\}$ modifies its adapted attention state, where 1 and 0 indicate whether token i is activated or not in the adapted token interaction pattern. 这里说是为了避免存储**连续的mask会占用存储空间**，但是。。。。。一个mask能占多少呢。。。

进一步地，基于二值的mask，有：

$$\tilde{\Psi}_h^{i,j}=\begin{cases}\frac{\exp\left(A_h^{i,j}\right)}{\sum_{s=1}^n m^s\exp\left(A_h^{i,s}\right)}, & \text{if } m^j=1;\\ 0, & \text{otherwise.}\end{cases}$$

在此基础上，遇到了一个新问题：**二值的mask在BP的时候没办法更新**。在这里我想到的方法是：**是否可以用一种连续的方法生成二值的mask呢？类似于HAT，基于一个embedding来生成mask**

在这里，作者采用了再参化技巧Gumbel softmax。对于第i个token的mask，作者定义了一个采样分布$t^i \in \mathbb{R}^2$，$t^{i,1}$表示mask的概率。然后使用Gumbel softmax将其转换成可导的形式：

$$m^i = \frac{exp((log(t^{i,1}) + g^1)/\tau)}{\sum_{k=1}^{2} exp((log(t^{i,k}) + g^k)/\tau)}$$

2.2 Attention to FFN

作者还为FFN设计了attention mask，对FFN的参数$W \in \mathbb{R}^{d_1 \times d_2}$的更新进行限制。

$$\tilde{w}^{i,j} = m^{i,j} w^{i,j} = \begin{cases} w^{i,j}, & \text{if } m^{i,j} = 1; \\ 0, & \text{otherwise.} \end{cases}$$

这里mask的生成跟上面一致。

2.3 Optimization Objective

除了一般用到的cross entropy loss之外，为了防止过多的token被mask掉造成严重的掉点，作者了另外介绍了一个drop control loss：

$$\mathcal{L}_{dc}(m) = \frac{1}{L} \sum_{l=1}^{L} \left(\lambda - \frac{1}{n} \sum_{i=1}^{n} m_l^i \right)^2 ,$$

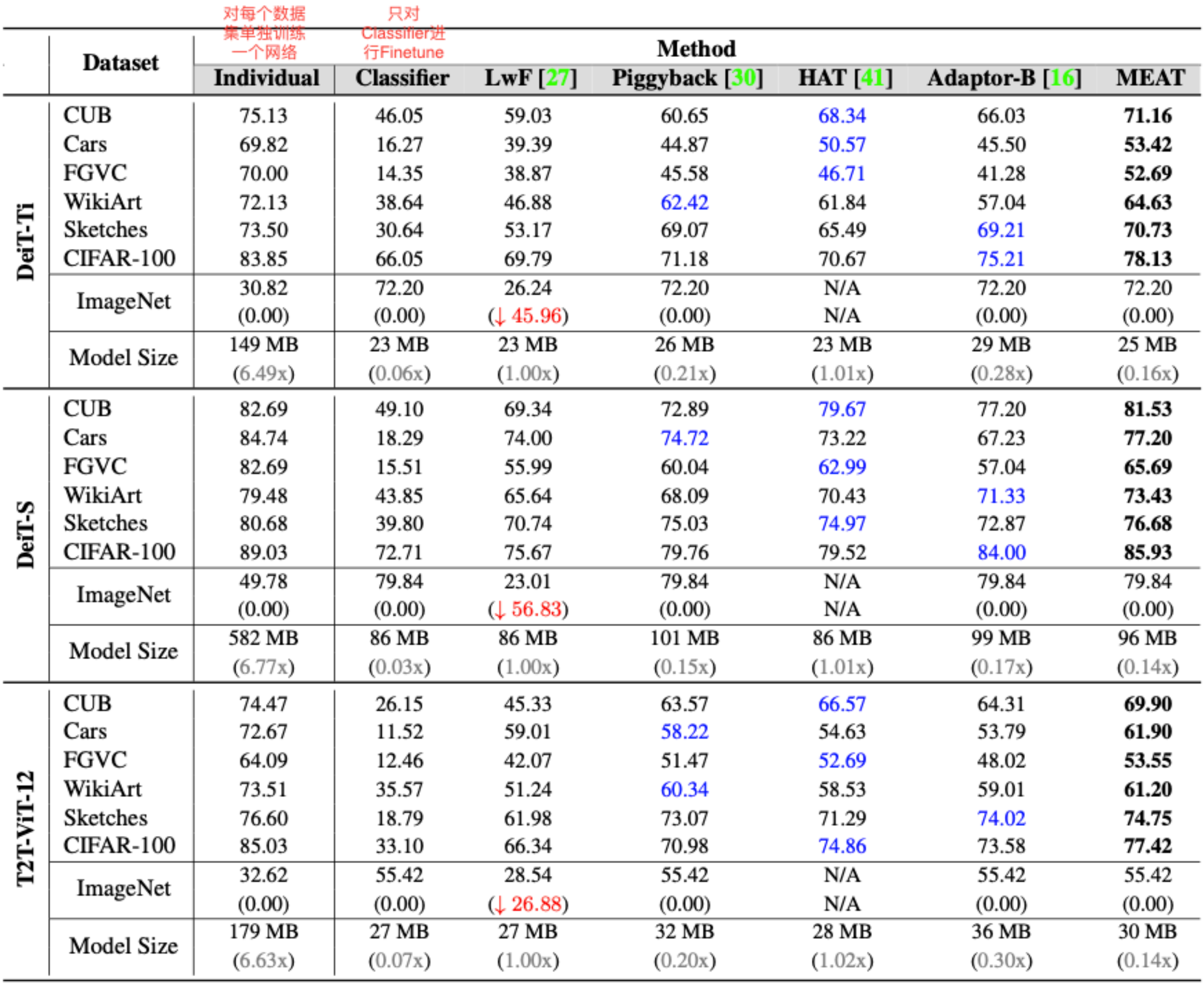
这里相当于是定义一个遮盖率来限制遮盖。

最后的loss写为：

$$L = L_{ce}(\hat{p}, p) + \alpha L_{dc}(m)$$

3、Experiment

3.1 Benchmark Comparison



对每个数据集单独训练一个网络

只对Classifier进行Finetune

	Dataset	Method						
		Individual	Classifier	LwF [27]	Piggyback [30]	HAT [41]	Adaptor-B [16]	MEAT
DeiT-Ti	CUB	75.13	46.05	59.03	60.65	68.34	66.03	**71.16**
	Cars	69.82	16.27	39.39	44.87	50.57	45.50	**53.42**
	FGVC	70.00	14.35	38.87	45.58	46.71	41.28	**52.69**
	WikiArt	72.13	38.64	46.88	62.42	61.84	57.04	**64.63**
	Sketches	73.50	30.64	53.17	69.07	65.49	69.21	**70.73**
	CIFAR-100	83.85	66.05	69.79	71.18	70.67	75.21	**78.13**
	ImageNet	30.82 (0.00)	72.20 (0.00)	26.24 (↓ 45.96)	72.20 (0.00)	N/A N/A	72.20 (0.00)	72.20 (0.00)
	Model Size	149 MB (6.49x)	23 MB (0.06x)	23 MB (1.00x)	26 MB (0.21x)	23 MB (1.01x)	29 MB (0.28x)	25 MB (0.16x)
DeiT-S	CUB	82.69	49.10	69.34	72.89	79.67	77.20	**81.53**
	Cars	84.74	18.29	74.00	74.72	73.22	67.23	**77.20**
	FGVC	82.69	15.51	55.99	60.04	62.99	57.04	**65.69**
	WikiArt	79.48	43.85	65.64	68.09	70.43	71.33	**73.43**
	Sketches	80.68	39.80	70.74	75.03	74.97	72.87	**76.68**
	CIFAR-100	89.03	72.71	75.67	79.76	79.52	84.00	**85.93**
	ImageNet	49.78 (0.00)	79.84 (0.00)	23.01 (↓ 56.83)	79.84 (0.00)	N/A N/A	79.84 (0.00)	79.84 (0.00)
	Model Size	582 MB (6.77x)	86 MB (0.03x)	86 MB (1.00x)	101 MB (0.15x)	86 MB (1.01x)	99 MB (0.17x)	96 MB (0.14x)
T2T-ViT-12	CUB	74.47	26.15	45.33	63.57	66.57	64.31	**69.90**
	Cars	72.67	11.52	59.01	58.22	54.63	53.79	**61.90**
	FGVC	64.09	12.46	42.07	51.47	52.69	48.02	**53.55**
	WikiArt	73.51	35.57	51.24	60.34	58.53	59.01	**61.20**
	Sketches	76.60	18.79	61.98	73.07	71.29	74.02	**74.75**
	CIFAR-100	85.03	33.10	66.34	70.98	74.86	73.58	**77.42**
	ImageNet	32.62 (0.00)	55.42 (0.00)	28.54 (↓ 26.88)	55.42 (0.00)	N/A N/A	55.42 (0.00)	55.42 (0.00)
	Model Size	179 MB (6.63x)	27 MB (0.07x)	27 MB (1.00x)	32 MB (0.20x)	28 MB (1.02x)	36 MB (0.30x)	30 MB (0.14x)

imageNet那一行有一点疑问，为什么其他方法在imageNet上效果不变，ViT backbone 是没有被更新吗?

3.2 Ablation study

3.2.1 Effectiveness of Components

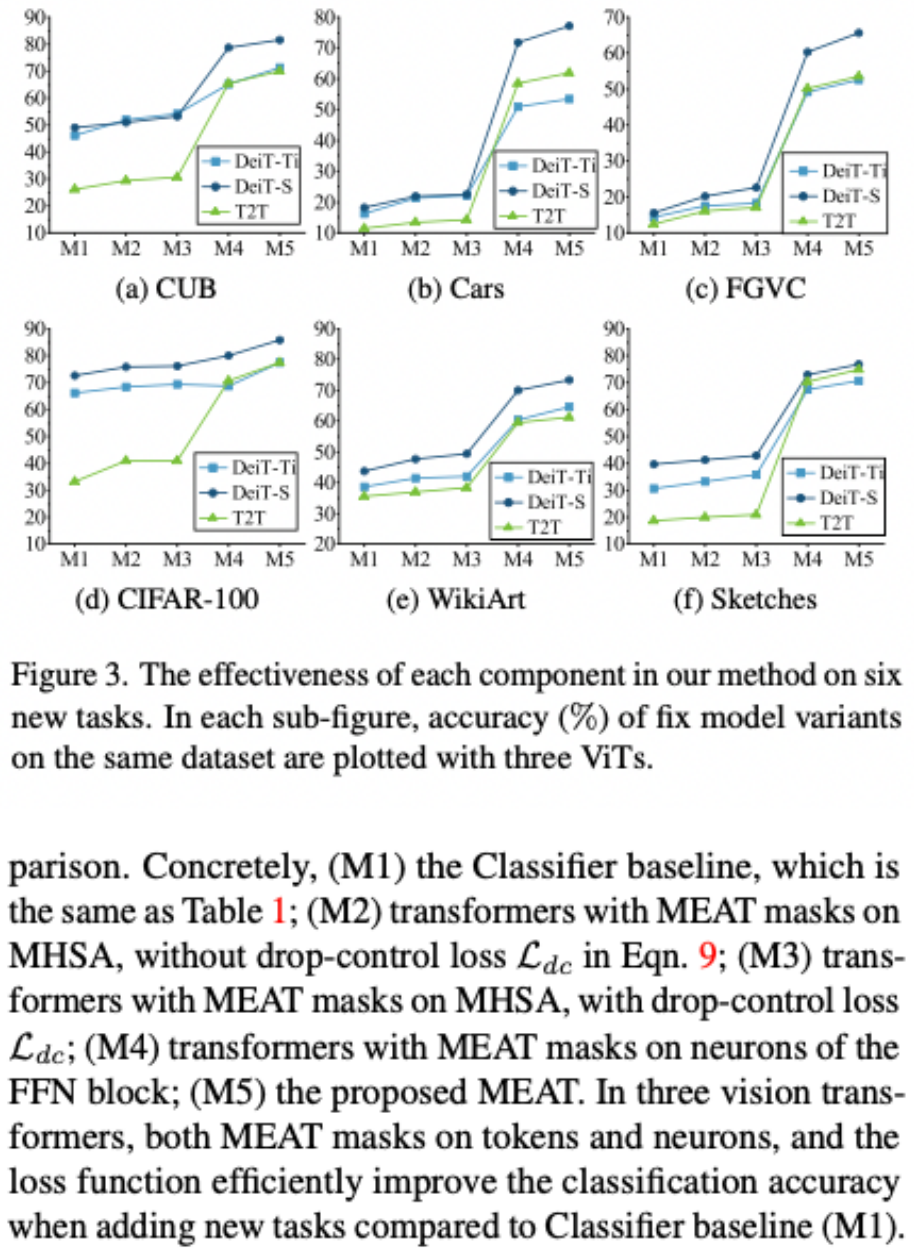


Figure 3. The effectiveness of each component in our method on six new tasks. In each sub-figure, accuracy (%) of fix model variants on the same dataset are plotted with three ViTs.

parison. Concretely, (M1) the Classifier baseline, which is the same as Table 1; (M2) transformers with MEAT masks on MHSA, without drop-control loss $\mathcal{L}_{dc}$ in Eqn. 9; (M3) transformers with MEAT masks on MHSA, with drop-control loss $\mathcal{L}_{dc}$; (M4) transformers with MEAT masks on neurons of the FFN block; (M5) the proposed MEAT. In three vision transformers, both MEAT masks on tokens and neurons, and the loss function efficiently improve the classification accuracy when adding new tasks compared to Classifier baseline (M1).

这么看起来，好像对FFN施加mask对效果影响最大?

3.2.2 Comparing to CNNs

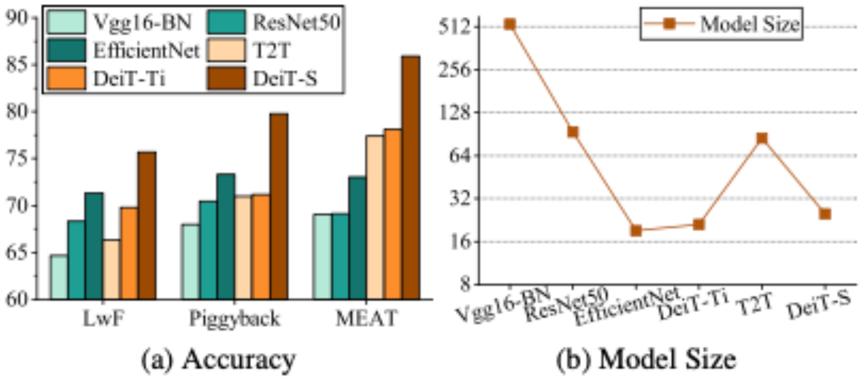


Figure 4. (a) Comparing results (%) over vision transformers and CNNs on CIFAR-100. (b) Model size (MB) comparison.

使用MEAT的CNNs架构相比其他方法只有一点提升，但是在Transformer上有很显著的增长。