Dropout Mixture Low-Rank Adaptation for Visual Parameters-Efficient Fine-Tuning (Supplementary Material)

Zhengyi Fang[1*], Yue Wang[1*], Ran Yi[1✉], and Lizhuang Ma[1,2]

[1] Shanghai Jiao Tong University
[2] MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University
{oliverfang,imwangyue,ranyi,lzma}@sjtu.edu.cn
https://github.com/Oliver7th/DMLoRA

1 Overview

This supplementary material mainly includes the following sections:

- Further analysis for DMLoRA training process, in Sec. 2.
- Imformation of pre-trained backbones, in Sec. 3
- Results on Swin-Transformer and ConvNeXt, in Sec. 4.
- Results on Self-Supervised pre-trained backbone, in Sec. 5.
- Few-shot learning results on FGVC, in Sec. 6.
- Investigation into the impact of dataset size on performance, in Sec. 7.
- More about training and memory cost, in Sec. 8.
- Hyper-parameters settings, in Sec. 9.
- More information about dataset, in Sec. 10.
- Implementation codes of our methods and experiments.

2 Further Analysis for DMLoRA Training Process

As the theoretical foundation for proposing DMLoRA, we believe that introducing a larger number of branches n, not only brings about strong regularization effects but also leads to a more dispersed parameter distribution among different branches. Consequently, the training of each branch is inevitably constrained by others. Through pruning, we can mitigate the mutual influence between branches. To explore the impact of branch numbers on training regularization and the degree of mutual restriction between different branches, we designed several sets of comparative experiments with different branch numbers (with no pruning through the process). Specifically, we saved the model parameters trained for 300 epochs on different type of representative datasets, and performed t-SNE dimensionality reduction on the weight parameters of the W_q

✉ Corresponding Author.
* Equal Contributions.

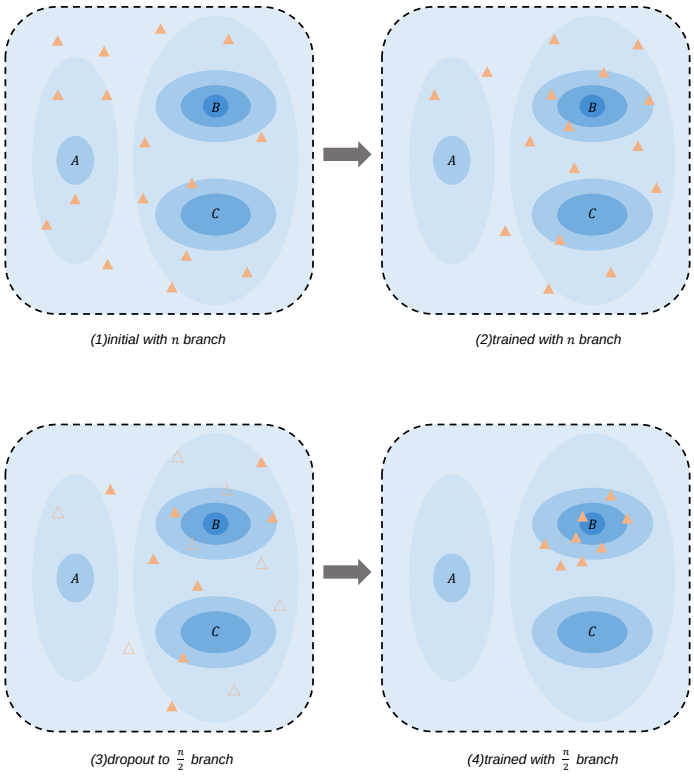

Fig. 1: Further analysis for DMLoRA training process.

of the last layer in the LoRA module. We report the range of the weights after dimensionality reduction in Tab. 1. The results indicate that the models with more branches exhibit larger range in the weight distributions of their MixLoRA branches, which proves that the greater the number of branches, the greater the dispersion of parameter distribution among branches. Based on these observed results, we can validate the effectiveness of branch pruning in DMLoRA.

The multi-branch pruning strategy of DMLoRA ensures that the model does not fall into poor local optima during the early stages of training, while stabilizing the model in the later stages of training. As shown in Fig. 1, during the early stages of training, the model has multiple branches, allowing it to choose from a more diverse set of training paths, preventing rapid convergence to poor local optimum A. In the later stages of training, several local optima of parameters may be closely adjacent. If the model maintains a high number of branches, the parameters may fluctuate between local optima B and C, leading to training instability and underperformance of the model. Reducing the number

Table 1: The range of two-dimensional coordinates of weight parameters after t-SNE dimensionality reduction. Results are represented by the sum of two dimensions.

	Natural			Specialized			Structured		
	Caltech101	Flowers102	SVHN	Camelyon	EuroSAT	Resisc45	Clevr-Dist	DMLab	sNORB-Ele
$n = 4$	210.7	166.1	302.4	813.4	104.4	198.6	503.9	755.8	619.2
$n = 8$	379.8	249.8	446.7	1141.2	160.8	277.7	579.7	1072.0	872.5
$n = 16$	516.2	323.8	529.4	1440.6	188.3	379.6	681.9	1127.5	939.2

Table 2: Specifications of different pre-trained backbones used in the paper. # Parameters (M) are of the feature extractor. All backbones are pre-trained on ImageNet with resolution 224×224

Backbone	Pre-trained Objective	Pre-trained Dataset	# params (M)	Feature Dim	Pre-trained Model
ViT-B/16 [2]	Supervised	ImageNet-21k	85.8	768	checkpoint
ViT-B/16 [2]	MoCo V3 [5]	ImageNet-1k	85.8	768	checkpoint
ViT-B/16 [2]	MAE [4]				checkpoint
Swin-B [10]	Supervised	ImageNet-21k	86.7	1024	checkpoint
ConvNeXt-B [11]	Supervised	ImageNet-21k	87.6	1024	checkpoint

of branches at the appropriate time can improve training stability and enhance model performance.

3 Pre-trained Backbone

This section primarily introduces the backbones used in the paper, including their structures, parameter counts, and pretraining methods, as depicted in Tab. 2. Specifically, we employed three network architectures: ViT [2], Swin-Transformer [10], and ConvNeXt [11], along with two pretraining methods: supervised learning and self-supervised learning.

4 Results on More Backbones

In this section, we validate the effectiveness of our method on Swin-Transformer-B [10] and ConvNeXt-B [11] architectures. For Swin-Transformer-B, we deployed DMLoRA-LS in the window attention layers, while for ConvNeXt-B, we incorporated DMLoRA into the first linear layer of the MLP within each block. We conducted experiments on VTAB-1K, and the results are presented in Tab. 3. It can be observed that: (1) Compared to ViT-B/16, both Swin-Transformer

Table 3: Results of **DMLoRA-LS** on different network architectures on VTAB-1k. "Avg" denotes the average accuracy. "Nat", "Spe" and "Str" are the average accuracies of the Natural, Specialized and Structured datasets, respectively. Top-1 accuracy (%) is reported. The best result is in **bold**, and the second-best result is underlined.

Model	Method	Avg.	Nat.	Spe.	Str.
Convolutional Network:					
ConvNeXt-B	Full	74.0	78.0	83.7	60.4
ConvNeXt-B	Linear	63.6	74.5	81.5	34.8
ConvNeXt-B	VPT [8]	68.7	78.5	83.0	44.6
ConvNeXt-B	LoRA [7]	72.1	79.2	83.4	53.8
ConvNeXt-B	RepAdapter [12]	79.0	83.5	86.7	**66.8**
ConvNeXt-B	**DMLoRA-LS (Ours)**	**79.3**	**84.4**	**88.0**	65.4
Hierarchical Vision Transformer:					
Swin-B	Full	75.0	79.2	86.2	59.7
Swin-B	Linear	62.6	73.5	80.8	33.5
Swin-B	VPT [8]	71.6	76.8	84.5	53.4
Swin-B	RepAdapter [12]	77.4	82.7	87.5	62.0
Swin-B	**DMLoRA-LS (Ours)**	**77.8**	**83.0**	**87.8**	**62.6**
Vision Transformer:					
ViT-B/16	Full	68.9	75.9	83.4	47.6
ViT-B/16	Linear	57.6	68.9	77.2	26.8
ViT-B/16	VPT [8]	72.0	78.5	82.4	55.0
ViT-B/16	RepAdapter [12]	76.0	81.6	85.4	61.2
ViT-B/16	**DMLoRA-LS (Ours)**	**77.0**	**82.5**	**86.3**	**62.1**

and ConvNeXt achieved superior performance. Specifically, the averate accuracy of Swin-Transformer outperformed ViT-B/16 by 0.5% in Natural, 1.5% in Specialized, and 0.5% in Structure datasets. For ConvNeXt, it outperformed ViT-B/16 by 2.3% in avearage accuracy. (2) Compared to other state-of-the-art fine-tuning algorithms, our method surpassed previous methods by 0.4% for Swin-Transformer and by 0.3% for ConvNeXt. These results demonstrate the wide applicability of our DMLoRA method across various backbones.

5 Results on Self-Supervised Pre-trained Backbone

In this section, we explore the performance of backbones obtained through different self-supervised pretraining methods, as illustrated in Tab. 4. We conducted experiments separately using two self-supervised pretraining methods: MAE and MoCoV3, with a ViT-B/16 backbone. It can be observed that: (1) The overall performance of backbones from these two self-supervised pretraining methods is inferior to those pretraining results achieved on the supervised ImageNet-22K dataset. However, there are exceptions in certain smaller-scale results. For instance, on the Structure dataset, where self-supervised pretraining methods (64.0%) outperform supervised pretraining methods with ViT-B/16 (62.1%) by 1.9%. (2) Compared to previous methods, our approach performs 2.6% better on

Table 4: Results of different self-supervised pre-trained backbones: MAE [4] and MoCo [5] with a ViT-B/16 backbone. Top-1 accuracy (%) is reported. The best result is in **bold**, and the second-best result is underlined.

	MAE				MoCo			
	Avg.	Nat.	Spe.	Str.	Avg.	Nat.	Spe.	Str.
Full Finetune	64.3	**59.3**	79.7	53.8	69.6	72.0	84.7	52.0
Linear	32.1	18.9	53.7	23.7	59.6	67.5	81.1	30.3
BitFit [18]	59.3	54.6	75.7	47.7	69.2	72.9	81.1	53.4
Adapter [6]	56.4	54.9	75.2	39.0	68.2	74.2	82.7	47.7
VPT-shallow [8]	45.7	40.0	69.7	27.5	62.4	67.3	82.3	37.6
VPT-deep [8]	41.1	36.0	60.6	26.6	65.2	70.3	83.0	42.4
DMLoRA-LS (Ours)	**66.9**	56.6	**82.6**	**61.4**	**75.0**	**74.9**	**86.2**	**64.0**

Table 5: Results of average accuracy on FGVC subsets, comparing LoRA and DMLoRA.

Method	10% FGVC	50% FGVC	80% FGVC	100% FGVC
DMLoRA	**72.1**	**88.0**	**90.0**	**90.7**
LoRA	68.9	87.2	89.4	90.2

MAE pre-trained backbone, 5.4% better on MoCo pre-trained backbone. These results indicate that our method is less influenced by the choice of pretraining methods and achieves relatively good results across various pretraining strategies.

6 Few-shot Learning on FGVC

In this section, we validate the performance of our method under the setting of few-shot learning. Specifically, following the setup of CoOp [20], we conducted experiments on the Few-shot FGVC dataset. To assess our method's performance under different numbers of shots, we conducted experiments using 1, 2, 4, 8, and 16-shot settings. We used the supervised pretraining ViT-B/16 for comparative experiments, and the results are illustrated in Fig. 2. It can be observed that our method shows an improvement in all settings compared to previous methods.

7 Investigation into the Impact of Dataset Size on Performance

To assess the impact of dataset size on algorithm performance, we conducted experiments to validate the performance degradation of the DMLoRA and LoRA on 10%, 50% and 80% of the FGVC training dataset. The results shown in Tab. 5, together with Fig. 2 and Table 1 and Table 2 in the main text of the paper, support a conclusion that the gap between different methods narrows as the

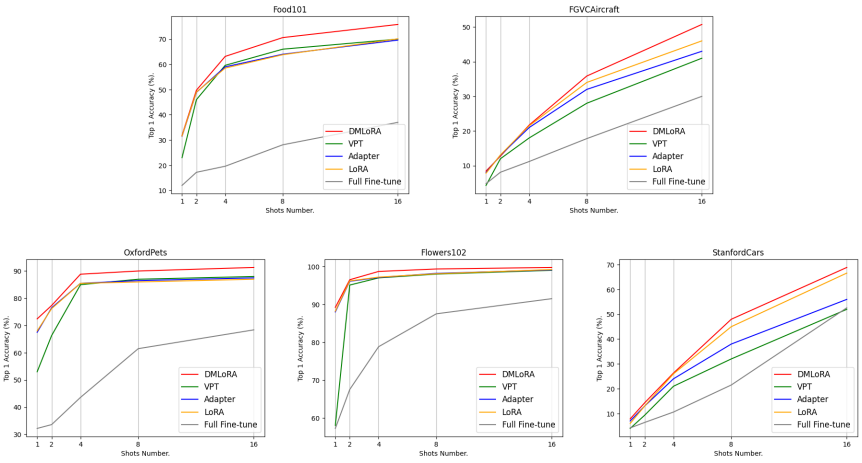

Fig. 2: Results on few-shot FGVC. Our method shows an improvement in all settings.

data scale increases. On the other hand, it demonstrates that DMLoRA exhibits less performance degradation with reduced training data, further substantiating the superiority of our algorithm.

8 Comparisons of Training and Memory Cost

To further demonstrate the efficiency of our method in training time and memory, we quantitatively compared DMLoRA with baseline LoRA [7] and the previous state-of-the-art method RepAdapter [12] through experiments.

With single NVIDIA RTX 4090 GPU, we conducted comparative experiments on VTAB-1K dataset, utilizing ViT-B/16 as backbone, batchsize = 64. For memory cost, DMLoRA requires only 12.33GB of VRAM, just 0.2% higher than LoRA, and saves 2.7% compared to RepAdapter. For training time, we compared the average pure training time (in seconds) required to achieve the best results on the VTAB-1k dataset by several methods. Compared to LoRA, we only used 2% more training time per epoch and about 20% more total training time to achieve a significant improvement in accuracy. Meanwhile, compared to the previous state-of-the-art method RepAdapter, we achieved higher accuracy with less training time. Detailed data comparison can be viewed in the Tab. 6.

9 Hyper-Parameters Settings

In this section, we introduce the hyperparameters utilized in our experiments. The crucial hyperparameters adjusted in our experiments include Learning Rate, Training Epoch, Initial Scale, Dropout Epoch, etc. With our proposed MixLoRA's

Table 6: The training time quantification experiment on the VTAB-1K dataset, where the per epoch column represents the time spent training one epoch of dataset, and the total column represents the time from the start of training to obtaining the best results

Method	Time(/sec)		acc(%)
	per epoch	total	
LoRA [7]	3.14	919.7	74.5
RepAdapter [12]	2.94	1318.8	75.4
DMLoRA-LS(**Ours**)	3.20	1161.9	77.0

structure, as the number of branches increases, the probability of each branch being selected and trained decreases. Due to this characteristic, more training epochs and lower learning rate are needed in order to fully train each branch. We appropriately increased the total number of training epochs and adjusted the learning rate based on the dataset's properties. The initial scale is select from $\{1, 10\}$, learning scale is selected from $\{2e-4, 5e-4, 1e-3\}$, the training epoch is select from $\{300, 600, 900\}$, the initial expert number is selected from $\{4, 6, 8\}$, and the dropout epoch is selected from $\{10, 20, 100, 200\}$. For more information, please refer to our codes.

Some patterns were revealed during experiments. In VTAB-1K, most of datasets performed best when initial branch number was set to 8. Regarding the dropout schedule, most of datasets in **VTAB Natural** and **VTAB Specialized** achieve better results with slow branch pruning at a frequency of 100/200 epochs each time(Caltech101 is an exception), whereas datasets in **VTAB Structured** perform better with a fast branch pruning strategy at a frequency of 10/20 epochs each time.

We conducted ablation studies to explore the impact of different hyperparameter settings on 3 types of datasets in VTAB-1K. From the results in table below, slow-pruning settings (**S1**, **S2**) perform significantly better in **Natural** and **Specialized**, while the fast-pruning (**S3**) setting has an advantage in **Structured**. And when the initial branch number decreases from 8 to 6 (**S1** to **S2**), the model's performance on various datasets shows a slight decline. In summary, it is reasonable to infer that for natural image datasets (Natural) and perception-oriented image datasets (Specialized) in specific domains, a larger dropout epoch should be set to help them find better optimization directions. For datasets that focus on logic, spatial awareness and orientation (Structured), a smaller dropout epoch should be used to facilitate rapid and stable convergence. Refer to Tab. 7 for detailed data.

Table 7: Ablation study on hyperparameters settings.

Settings	**Natural**	**Specialized**	**Structured**	Avg
S1 (branch=8, drop_ep=100)	<u>82.0</u>	<u>86.2</u>	60.9	<u>76.4</u>
S2 (branch=6, drop_ep=100)	81.9	86.1	60.8	76.3
S3 (branch=8, drop_ep=10)	80.5	85.8	<u>61.7</u>	76.0
Optimal settings (Ours)	**82.5**	**86.3**	**62.0**	**77.0**

10 More Information about Dataset

In this paper, we utilize three datasets: VTAB-1K [19], FGVC, Few-shot FGVC. We summarized the training, testing, and validation data quantities of these datasets in Tab. 8. These datasets contain many subsets, some of which overlap. Despite sharing identical subset names, they employ different training data quantities, thereby constituting distinct tasks. VTAB-1K comprises 1,000 images per subset, FGVC utilizes the native training datasets, and Few-shot FGVC employs 1 to 16 training images per set.

Table 8: Specifications of the various datasets evaluated.

Dataset	Description	# Classes	Train	Val	Test
Visual Task Adaptation Benchmark (VTAB-1K) [19]					
CIFAR-100	Natural	100	800/1000	200	10,000
Caltech101		102			6,084
DTD		47			1,880
Flowers102		102			6,149
Pets		37			3,669
SVHN		10			26,032
Sun397		397			21,750
Patch Camelyon	Specialized	2	800/1000	200	32,768
EuroSAT		10			5,400
Resisc45		45			6,300
Retinopathy		5			42,670
Clevr/count	Structured	8	800/1000	200	15,000
Clevr/distance		6			15,000
DMLab		6			22,735
KITTI/distance		4			711
dSprites/location		16			73,728
dSprites/orientation		16			73,728
SmallNORB/azimuth		18			12,150
SmallNORB/elevation		9			12,150
Fine-grained visual recognition tasks (FGVC)					
CUB-200-2011 [17]	Fine-grained bird species recognition	200	5,394	600	5,794
NABirds [16]	Fine-grained bird species recognition	555	21,536	2,393	24,633
Oxford-flowers102 [14]	Fine-grained flower species recognition	102	1,020	1,020	6,149
Stanford-Dogs [9]	Fine-grained dog species recognition	120	10,800	1,200	8,580
Stanford-Cars [3]	Fine-grained car classification	196	7,329	815	8,041
Few-shot Fine-grained visual recognition tasks (Few-shot FGVC)					
Food-101 [1]	Fine-grained food classification	101	1/2/4/8/16 per class	20,200	30,300
FGVC-Aircraft [13]	Fine-grained aircraft classification	100		3,333	3,333
Oxford-Flowers102 [14]	Fine-grained flower species recognition	102		1,633	2,463
Oxford-Pets [15]	Fine-grained pets species recognition	37		736	3,669
Stanford-Cars [3]	Fine-grained car classification	196		1,635	8,041

References

1. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13. pp. 446–461. Springer (2014) 8
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 3
3. Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., Fei-Fei, L.: Fine-grained car detection for visual census estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 31 (2017) 8
4. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022) 3, 5
5. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020) 3, 5
6. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: International Conference on Machine Learning. pp. 2790–2799. PMLR (2019) 5
7. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net (2022), https://openreview.net/forum?id=nZeVKeeFYf9 4, 6, 7
8. Jia, M., Tang, L., Chen, B., Cardie, C., Belongie, S.J., Hariharan, B., Lim, S.: Visual prompt tuning. In: Avidan, S., Brostow, G.J., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIII. Lecture Notes in Computer Science, vol. 13693, pp. 709–727. Springer (2022). https://doi.org/10.1007/978-3-031-19827-4_41, https://doi.org/10.1007/978-3-031-19827-4_41 4, 5
9. Khosla, A., Jayadevaprakash, N., Yao, B., Li, F.F.: Novel dataset for fine-grained image categorization: Stanford dogs. In: Proc. CVPR workshop on fine-grained visual categorization (FGVC). vol. 2. Citeseer (2011) 8
10. 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) 3
11. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022) 3
12. Luo, G., Huang, M., Zhou, Y., Sun, X., Jiang, G., Wang, Z., Ji, R.: Towards efficient visual adaption via structural re-parameterization. CoRR **abs/2302.08106** (2023). https://doi.org/10.48550/ARXIV.2302.08106, https://doi.org/10.48550/arXiv.2302.08106 4, 6, 7

13. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013) 8
14. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian conference on computer vision, graphics & image processing. pp. 722–729. IEEE (2008) 8
15. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3498–3505. IEEE (2012) 8
16. Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., Belongie, S.: Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 595–604 (2015) 8
17. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011) 8
18. Zaken, E.B., Ravfogel, S., Goldberg, Y.: Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. arXiv preprint arXiv:2106.10199 (2021) 5
19. Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A.S., Neumann, M., Dosovitskiy, A., et al.: A large-scale study of representation learning with the visual task adaptation benchmark. arXiv preprint arXiv:1910.04867 (2019) 8
20. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16816–16825 (2022) 5