Supplementary Material for "PYRA: Parallel Yielding Re-Activation for Training-Inference Efficient Task Adaptation"

Yizhe Xiong[1,2], Hui Chen[2*], Tianxiang Hao[1,2], Zijia Lin[1], Jungong Han[2,3], Yuesong Zhang[4], Guoxin Wang[4], Yongjun Bao[4], and Guiguang Ding[1,2]

[1] School of Software, Tsinghua University, Beijing, China
[2] BNRist, Tsinghua University, Beijing, China
[3] Department of Automation, Tsinghua University, Beijing, China
[4] JD.com, Beijing, China
{xiongyizhe2001,beyondhtx,jungonghan77}@gmail.com
huichen@mail.tsinghua.edu.cn
{zhangyuesong1,wangguoxin14,baoyongjun}@jd.com
linzijia07@tsinghua.org.cn dinggg@tsinghua.edu.cn

Table 1: The statistics of the VTAB-1k [37] benchmark.

Dataset	Description	Classes	Train size	Val size	Test size
CIFAR-100 [24]	Natural	100	800/1000	200	10000
Caltech101 [11]		102			6084
DTD [7]		47			1880
Flowers102 [31]		102			6149
Pets [32]		37			3669
SVHN [30]		10			26032
Sun397 [36]		397			21750
Patch Camelyon [34]	Specialized	2	800/1000	200	32768
EuroSAT [16]		10			5400
Resisc45 [6]		45			6300
Retinopathy [13]		5			42670
Clevr/count [22]	Structured	8	800/1000	200	15000
Clevr/distance [22]		6			15000
DMLab [2]		6			22735
KITTI/distance [12]		4			711
dSprites/location [29]		16			73728
dSprites/orientation [29]		16			73728
SmallNORB/azimuth [25]		18			12150
SmallNORB/elevation [25]		9			12150

* Corresponding Author

Table 2: PEFT hyperparameters for each tested backbone.

	ViT-B/16	ViT-L/16	ViT-L/16 (MAE)	DeiT-B/16
LoRA h	8	12	12	8

Table 3: Training hyperparameters for each tested backbone.

	ViT-B/16	ViT-L/16	ViT-L/16 (MAE)	DeiT-B/16
optimizer	AdamW	AdamW	AdamW	AdamW
warmup epochs	10	10	10	10
epochs	100	100	100	100
batch size	64	32	32	128
lr (PEFT)	1e-3	1e-3	1e-3	1e-3
wd	1e-4	1e-4	1e-4	1e-4

A Details for the Evaluation Datasets

We show the detailed statistics of the Visual Task Adaptation Benchmark (VTAB-1k) [37] in Tab. 1. Introduced in [37], the VTAB-1k benchmark contains 19 tasks from diverse domains: (1) *Natural* images that are captured by standard cameras in real-world scenarios; (2) *Specialized* images that are captured by professional equipment, *e.g.*, remote sensing and medical cameras; (3) *Structured* images that are synthesized from simulated environments. VTAB-1k contains a variety of tasks such as object counting and depth estimation apart from the standard image classification. For each task in VTAB-1k, the images are divided into the training set (800 images), the validation set (200 images), and the test set (the original set). All models are fine-tuned on the train+val set, *i.e.*, 1000 images. The validation set is used for hyperparameter tuning.

B More Implementation Details

We present details of implementing PYRA for training-inference efficient task adaptation.

PEFT Hyperparameters. For training-inference efficient task adaptation, we implement LoRA [19] with hidden layer dimension h as the PEFT module regarding its simplicity and mergeability. We present the PEFT hyperparameters in Tab. 2. Note that h choices in Tab. 2 lead to approximately the same percentage of training parameters compared to the backbone for all vision transformers. We use the same h value for each backbone under all competing methods and our PYRA.

Training Hyperparameters. For choices of training hyperparameters, we mainly follow [14,38] and only conduct minor adjustments on the training batch size for efficient training on our devices. The hyperparameters are listed in Tab. 3. To prevent the token modulation from overly altering tokens, thereby leading to training instability, we adopt different learning rates for the modulation weight

Table 4: The learning rate choices for W_D and W_r in each tested backbone.

	lr choices for W_D and W_r
ViT-B/16	[1e-5, 3e-5, 1e-4, 3e-4, 1e-3, 3e-3]
ViT-L/16	[1e-5, 3e-5, 1e-4, 3e-4, 1e-3, 3e-3]
ViT-L/16 (MAE)	[1e-6, 3e-6, 1e-5, 3e-5]
DeiT-B/16	[1e-5, 3e-5, 1e-4, 3e-4, 1e-3, 3e-3]

Table 5: The merging schedules for the high compression rate benchmark.

	r schedule ($[r^1, r^2, \cdots, r^L]$)
ViT-B/16	[40,34,30,24,18,14,10,8,4,4,3,3]
ViT-L/16	[20,19,18,17,15,13,13,12,10,9,8,6,6,4,4,4,3,3,2,2,1,1,1,1]
ViT-L/16 (MAE)	[20,19,18,17,15,13,13,12,10,9,8,6,6,4,4,4,3,3,2,2,1,1,1,1]
DeiT-B/16	[40,34,30,24,18,14,10,8,4,4,3,3]

generators W_D and W_r introduced in PYRA. Specifically, for each tested model, we tune for the optimal learning rate for W_D and W_r on each task within the corresponding range as listed in Tab. 4.

Changing Compression Rate for PYRA. To yield compressed smaller-scale models with different throughput values, we change the compression rate of PYRA by changing the r value, *i.e.*, merged token number, for each layer.

(1) For the low compression rate benchmark, we simply implement a constant merging schedule which simply merges the same number of tokens for each layer. Specifically, we merge 16 tokens in each layer for ViT-B/16 and DeiT-B/16, and merge 8 tokens in each layer for ViT-L/16 and ViT-L/16 (MAE).

(2) For the high compression rate benchmark, we implement a decreasing schedule that consistently outperforms other schedules on the pre-trained dataset [3]. To achieve the target throughput value demanded by the high compression rate benchmark, we solve an approximate problem and apply minor adjustments to the solutions based on actually tested throughput values. Formally, for a vision transformer with L layers, total token number T (here we omit the [CLS] token and the distillation token), and F computational FLOPs, we compress it to a smaller-scale model with f FLOPs by merging r^l tokens in layer l. We denote the number of remaining tokens in the last layer as t (usually $t < 5$). We calculate the approximate solution $\hat{r}^l$ for the merging schedule as:

$$\hat{r}^l = \lfloor (g(l-1) - g(l))(T-t) \rfloor, \text{where } g(x) = (1 - \frac{x}{L})^{\frac{F}{f}-1}. \tag{1}$$

After acquiring the approximate solutions $\hat{r}^l$, we apply minor adjustments based on the actually tested throughput values to yield the demanded speedup. The final adjusted r^l yields a smaller-scale compressed model with similar throughput values as the smaller-scale backbone, as listed in Tab. 3 of the article. The actually applied r^l values for the high compression rate benchmark are listed in Tab. 5. Note that Eq. (1) can also yield the adopted constant merging schedule for the low compression rate benchmark.

Table 6: Comparison of different token pruning method choices on the ViT-B/16 with LoRA modules attached as the PEFT choice.

Compression Rate	Method	# params	Throughput	Average
Low	EViT [27]	0.34%	732	73.09
	ToFu [23]	0.34%	748	73.31
	ATS [10]	0.34%	727	71.16
	LTMP [4]	0.34%	724	52.70
	ToMe [3]	0.34%	753	<u>74.10</u>
	PYRA	0.35%	745	**74.69**
High	EViT [27]	0.34%	1200	65.79
	ToFu [23]	0.34%	1370	70.39
	ATS [10]	0.34%	1183	61.12
	LTMP [4]	0.34%	1265	48.96
	ToMe [3]	0.34%	1381	<u>70.43</u>
	PYRA	0.35%	1365	**72.06**

Table 7: Comparison to PE-Dist under the high compression rate.

Backbone	Method	# params	Throughput	Natural	Specialized	Structured	Average
ViT-S/16	PEFT	0.34%	1350	**76.29**	<u>83.56</u>	<u>55.71</u>	<u>71.85</u>
	PE-Dist	0.34%	1350	**76.29**	**83.84**	52.85	70.99
ViT-B/16	PYRA	0.35%	1365	<u>73.91</u>	82.60	**59.66**	**72.06**
ViT-B/16	PEFT	0.34%	425	79.45	<u>84.43</u>	<u>60.39</u>	<u>74.76</u>
	PE-Dist	0.34%	425	<u>80.30</u>	84.24	59.26	74.60
ViT-L/16	PYRA	0.40%	427	**80.43**	**85.17**	**61.39**	**75.66**

For benchmark results in Sec. 4.2 of the article, we adopt the same merging schedules for PYRA and ToMe [3] to guarantee a fair comparison.

C Comparing to More Baseline Methods

Apart from the baseline methods chosen in the main article, we applied several more token pruning methods to demonstrate that our choice of ToMe [3] is a strong baseline in the setting of training-inference efficient task adaptation. Specifically, we have applied EViT [27], ToFu [23], ATS [10], and LTMP [4] with LoRA attached as the learnable PEFT module. As shown in Tab. 6, all listed methods underperform the strongest baseline, ToMe [3], well demonstrating that analyzing and evaluating ToMe in training-inference efficient task adaptation is important and appropriate. Note that although [4, 23] are follow-up works of ToMe, they involve heavy token pruning that requires extensive training to restore performance, hence underperforming ToMe in PEFT training.

D Comparing to Parameter-Efficient Model Distillation

Model distillation [17] is an alternative approach for acquiring a smaller-scale model of the desired size. Existing approaches to model distillation [1, 17, 33]

Table 8: Comparison of different pipelines on the ViT-B/16.

Compression Rate	Pipeline	Method	# params	Throughput	Average
Low	One-Stage	ToMe	0.34%	753	**74.10**
	Two-Stage		0.34%	753	72.73
High	One-Stage	ToMe	0.34%	1381	**70.43**
	Two-Stage		0.34%	1381	69.25

require training all parameters in the smaller-scale model with a combined loss:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{dist}}, \tag{2}$$

where $\mathcal{L}_{\text{task}}$ represents the task-specific loss, and $\mathcal{L}_{\text{dist}}$ denotes the distillation loss, typically measured by the KL-divergence.

In the setting of training-inference efficient task adaptation, however, training all parameters in the smaller-scale model is unacceptable. Therefore, we implement a baseline as *parameter-efficient model distillation* (PE-Dist), in which we *attach PEFT modules to existing pre-trained smaller-scale backbones, and only train the PEFT modules during model distillation.* We conduct experiments on the ViT backbones [9], *i.e.*, ViT-B/16 and ViT-L/16, by distilling them to the smaller-scale backbone, *i.e.*, ViT-S/16 and ViT-B/16, correspondingly. To yield the distillation source, we first train ViT-B/16 and ViT-L/16 by PEFT. As a fair comparison, we choose LoRA [19] as the PEFT method and compare the results for PE-Dist with our PYRA under the high compression rate. As shown in Tab. 7, parameter-efficient model distillation consistently underperforms directly fine-tuning the smaller-scale model for both backbones, indicating that the *adverse compression* phenomenon still exists when applying model distillation for training-inference efficient task adaptation. In contrast to model distillation, our PYRA consistently eliminates the adverse compression gap and outperforms model distillation by over $>1\%$ on both backbones as well. This indicates that PYRA represents an effective alternative to model distillation for acquiring smaller-scale inference-efficient models under the constraints of parameter-efficient training.

E More Analysis Experiments.

Is a Two-Stage Schedule Better? As discussed in Sec. 3.1 in the article, we employ a one-stage pipeline in which LoRA modules are trained while token merging is attached to the vision transformer. In fact, a more straightforward approach is to implement in a two-stage pipeline: we train the LoRA modules without token merging first, and then employ token merging in the second stage to reduce the redundant tokens. To compare the performance of both pipelines, we apply the ViT-B/16 model pre-trained on ImageNet-21K [8] with ToMe attached as the compression method. As shown in Tab. 8, under both low and high compression rates, the one-stage pipeline described in the main article outperforms the two-stage pipeline, well demonstrating that our pipeline choice is appropriate.

Table 9: Adaptation performance comparison between conducting token modulation via PYRA and simply increasing the training parameters in LoRA for the baseline.

Method	h	# params	Δ params	Natural	Specialized	Structured	Average	Δ Acc.
Baseline	8	0.34%	1.00×	72.9	80.8	57.6	70.43	0
Baseline	16	0.69%	2.00×	72.8	80.8	58.2	70.63	+0.20
Baseline	32	1.37%	4.00×	72.9	81.9	58.3	71.00	+0.57
PYRA	8	0.35%	1.03×	**73.9**	**82.6**	**59.7**	**72.06**	+1.63

Impact of Adding Trainable Parameters. To verify that the performance gain of PYRA does not come from the increase of training parameters, we compare PYRA with simply increasing the hidden dimension h of LoRA modules in the baseline. As shown in Tab. 9, the baseline method with increased h could only lead to minor performance gains. With 4× training parameters, the baseline still underperforms PYRA with only 1.03× training parameters by over 1.0%. These results well demonstrate the effectiveness of designs in PYRA.

Qualitative Analysis. To qualitatively analyze the effectiveness of our PYRA, we visualize the [CLS] token feature extracted by the strongest ToMe [3] baseline and our PYRA under high compression rate via t-SNE [28]. As shown in Fig. 1, PYRA yields features with clearer clusters compared to ToMe, demonstrating that PYRA can construct a better feature space for downstream tasks.

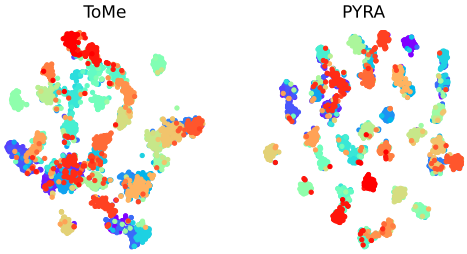

Fig. 1: t-SNE [28] visualization of ToMe [3] and PYRA on the Oxford Pet [32] dataset.

Modulation Matrix Rank. In PYRA, we insert a pair of modulation weight generators $W_r \in \mathbb{R}^{r\times 1}$ and $W_D \in \mathbb{R}^{1\times D}$ for each layer. To prove that setting the matrix rank s of both generators as 1 is enough, we increase the rank of generators and create PYRA variants with more training parameters for comparisons. To apply modulation weights with $s > 1$ to the merged tokens, we first calculate $W^l = \delta_D \delta_r$, and then apply token modulation by omitting Eq. 6 of the article and replacing $\hat{\delta}_r^l$ in Eq. 7 of the article to W^l. We compare our PYRA ($s = 1$) with other variants, *i.e.*, $s = 2$, $s = 4$, and $s = 8$ following Sec. 4.4 in the article, where we apply the ViT-B/16 model pre-trained on ImageNet-21K [8] under the high compression rate. As shown in Fig. 2, although introducing more training parameters, increasing s yields performance comparable to our PYRA with $s = 1$. This indicates that

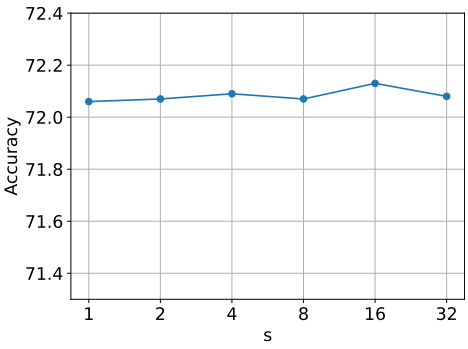

Fig. 2: Comparison of different modulation weight generator rank s in PYRA.

Table 10: Adaptation results for ViT-Base/16 under high compression rate with the Adapter [18] method. ∗: As a comparison of similar throughputs, we compare ViT-Small/16 with PEFT attached.

Method	# params	Throughput	Natural	Specialized	Structured	Average
PEFT∗	0.19%	1228	72.74	83.79	55.93	70.82
RaP [26]	0.70%	989	32.31	69.52	25.94	42.59
DiffRate [5]	0.19%	1244	48.18	70.55	23.67	47.46
ToMe [3]	0.18%	1301	72.37	80.57	56.06	69.66
PYRA	0.19%	1268	**74.07**	**82.25**	**57.46**	**71.26**

Table 11: Adaptation results for ViT-Large/16 under high compression rate with the Adapter [18] method. ∗: As a comparison of similar throughputs, we compare ViT-Base/16 with PEFT attached.

Method	# params	Throughput	Natural	Specialized	Structured	Average
PEFT∗	0.18%	393	79.33	84.69	57.41	73.81
RaP [26]	0.46%	240	20.24	62.89	30.14	37.76
DiffRate [5]	0.20%	391	49.60	45.79	22.47	39.29
ToMe [3]	0.20%	410	78.84	83.63	56.90	73.12
PYRA	0.21%	402	**79.39**	**84.31**	**58.00**	**73.90**

Table 12: Comparison of different PEFT method choices on the ViT-B/16. Results on ToMe and PYRA are reported.

Compression Rate	PEFT choice	ToMe	PYRA
Low	Convpass [20]	75.05	76.10
	FacT [21]	74.24	75.34
	SNF [35]	62.65	65.25
High	Convpass [20]	72.09	73.26
	FacT [21]	71.48	72.48
	SNF [35]	58.01	62.49

simply applying $W_r \in \mathbb{R}^{r\times 1}$ and $W_D \in \mathbb{R}^{1\times D}$ is effective and training-efficient to generate adaptive modulation weights that optimally modulate the tokens to be merged.

PYRA on other PEFT methods. For training-inference efficient task adaptation, we have mainly implemented LoRA [19] for its simplicity and mergeability. In conditions where the inference speed is not strictly restrained, PEFT methods that cannot merge into the backbone are also applicable in training-inference efficient task adaptation. To further validate the effectiveness of PYRA on other PEFT methods, we implement Adapter [18] as the PEFT method, which inevitably introduces small computational overhead during inference. We validate our PYRA on the ViT backbones [9] under the high compression rate. As shown in Tab. 10 and Tab. 11, PYRA achieves the best overall performance and outperforms all competing methods in each category. While surpassing the throughput of the smaller-scale backbone with only ∼0.2% training parameters, our PYRA eliminates the adverse compression gap between the compressed larger-scale backbone and the smaller-scale backbone. This indicates that our PYRA well generalizes to other PEFT methods in terms of achieving training-inference efficient task adaptation. Experiments on more recent PEFT methods listed in Tab. 12 also demonstrate that PYRA consistently outperforms the strongest baseline, ToMe [3], under a broad range of PEFT approaches.

Table 13: Performance comparison on the ImageNet-1k dataset [8]. "Plain" denotes directly fine-tuning the MAE pre-trained backbone on ImageNet-1k as in [15].

Model	Method	# Param	Throughput	Speedup	Acc.	Δ Acc.
ViT-Base/16 (MAE)	Plain [15]	86M	425	1.0×	83.66	0
	ToMe [3]	86M	1381	3.2×	76.67	-6.99
	PYRA	86M+9.4K	1365	3.2×	77.11	-6.55
ViT-Large/16 (MAE)	Plain [15]	303M	130	1.0×	85.95	0
	ToMe [3]	303M	431	3.3×	81.61	-4.34
	PYRA	303M+25K	427	3.3×	82.36	-3.59

F PYRA Performance on the Pre-Training Task

PYRA maintains discriminative information with adaptive token modulation, leading to improved performance while achieving complexity reduction via token merging. Extensive experiments have shown the effectiveness of PYRA in the challenge of training-inference efficient task adaptation. Apart from the task adaptation scenario, PYRA is also applicable to the pre-trained model for improving the performance of compressed backbones on the pre-training task.

We evaluate the effectiveness of PYRA on the pre-training task. Specifically, we follow [3] and apply PYRA during the fine-tuning process of self-supervised MAE backbones [15], *i.e.*, ViT-Base/16 (MAE) and ViT-Large/16 (MAE). We evaluate our PYRA under the challenging high compression rate. During fine-tuning, we simply tune the modulation weight generators with other trainable parameters under the same hyperparameters following [15]. The results are shown in Tab. 13. Under the high compression rate, ToMe [3] introduces significant performance drops. With $\sim$0.01% extra training parameters compared to the backbone, our PYRA significantly compensates the accuracy loss with adaptive token modulation. This can be attributed to that although our PYRA is specially designed to enhance the feature distribution in downstream tasks, it still brings improvements to the pre-training task where full fine-tuning is demanded, which indicates that the insight of improving the perception of feature distribution via token modulation inside our PYRA can be widely adopted to various scenarios.

References

1. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9163–9171 (2019)
2. Beattie, C., Leibo, J.Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., et al.: Deepmind lab. arXiv preprint arXiv:1612.03801 (2016)
3. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. arXiv preprint arXiv:2210.09461 (2022)
4. Bonnaerens, M., Dambre, J.: Learned thresholds token merging and pruning for vision transformers. arXiv preprint arXiv:2307.10780 (2023)

5. Chen, M., Shao, W., Xu, P., Lin, M., Zhang, K., Chao, F., Ji, R., Qiao, Y., Luo, P.: Diffrate: Differentiable compression rate for efficient vision transformers. arXiv preprint arXiv:2305.17997 (2023)
6. Cheng, G., Han, J., Lu, X.: Remote sensing image scene classification: Benchmark and state of the art. Proceedings of the IEEE **105**(10), 1865–1883 (2017)
7. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3606–3613 (2014)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. 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)
10. Fayyaz, M., Koohpayegani, S.A., Jafari, F.R., Sengupta, S., Joze, H.R.V., Sommerlade, E., Pirsiavash, H., Gall, J.: Adaptive token sampling for efficient vision transformers. In: European Conference on Computer Vision. pp. 396–414. Springer (2022)
11. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence **28**(4), 594–611 (2006)
12. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
13. Graham, B.: Kaggle diabetic retinopathy detection competition report. University of Warwick **22**, 17 (2015)
14. Hao, T., Chen, H., Guo, Y., Ding, G.: Consolidator: Mergeable adapter with grouped connections for visual adaptation. arXiv preprint arXiv:2305.00603 (2023)
15. 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)
16. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **12**(7), 2217–2226 (2019)
17. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
18. 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)
19. 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. arXiv preprint arXiv:2106.09685 (2021)
20. Jie, S., Deng, Z.H.: Convolutional bypasses are better vision transformer adapters. arXiv preprint arXiv:2207.07039 (2022)
21. Jie, S., Deng, Z.H.: Fact: Factor-tuning for lightweight adaptation on vision transformer. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 1060–1068 (2023)
22. Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2901–2910 (2017)

23. Kim, M., Gao, S., Hsu, Y.C., Shen, Y., Jin, H.: Token fusion: Bridging the gap between token pruning and token merging. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1383–1392 (2024)
24. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto (2009)
25. LeCun, Y., Huang, F.J., Bottou, L.: Learning methods for generic object recognition with invariance to pose and lighting. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol. 2, pp. II–104. IEEE (2004)
26. Li, Y., Adamczewski, K., Li, W., Gu, S., Timofte, R., Gool, L.V.: Revisiting random channel pruning for neural network compression. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 191–201. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.00029, https://doi.org/10.1109/CVPR52688.2022.00029
27. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Not all patches are what you need: Expediting vision transformers via token reorganizations. arXiv preprint arXiv:2202.07800 (2022)
28. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
29. Matthey, L., Higgins, I., Hassabis, D., Lerchner, A.: dsprites: Disentanglement testing sprites dataset (2017)
30. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y., et al.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning. vol. 2011, p. 7. Granada, Spain (2011)
31. 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)
32. 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)
33. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1365–1374 (2019)
34. Veeling, B.S., Linmans, J., Winkens, J., Cohen, T., Welling, M.: Rotation equivariant cnns for digital pathology. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11. pp. 210–218. Springer (2018)
35. Wang, Y., Shi, B., Zhang, X., Li, J., Liu, Y., Dai, W., Li, C., Xiong, H., Tian, Q.: Adapting shortcut with normalizing flow: An efficient tuning framework for visual recognition. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15965–15974. IEEE (2023)
36. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 3485–3492. IEEE (2010)
37. 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)
38. Zhang, Y., Zhou, K., Liu, Z.: Neural prompt search. arXiv preprint arXiv:2206.04673 (2022)