Supplementary Material for Sensitivity-Aware Visual Parameter-Efficient Fine-Tuning

Haoyu He[1] Jianfei Cai[1] Jing Zhang[2] Dacheng Tao[2] Bohan Zhuang[1†]

[1] Monash University [2] The University of Sydney

We organize our supplementary material as follows.

- In Section A, we introduce more details about the contenders.
- In Section B, we show more sensitivity patterns for ViT-B/16 with various pre-training strategies.
- In Section C, we show some dataset samples from ImageNet [12] and VTAB-1k [15].
- In Tables I and II, we show per-task results for our SPT variants on FGVC and VTAB-1k benchmarks, respectively.

A. More Details of Contenders

- FULL: fully tunes all the backbone and classification head parameters.
- LINEAR: freezes all the backbone parameters and only tunes a linear classification head.
- BIAS [14]: freezes all the backbone parameters except for the bias terms and also tunes the linear classification head.
- PARTIAL-k: freezes all the backbone parameters except for the last k layers and also tunes the linear classification head as described in [10].
- MLP-k: freezes all the backbone parameters and tunes the classification head which is implemented by a trainable k-layer multi-layer perceptron as described in [10].
- PROMPT-SHALLOW [10]: freezes all the backbone parameters while introducing additional trainable prompts to the input space of the pretrained ViT.
- PROMPT-DEEP [10]: freezes all the backbone parameters while appending additional trainable prompts to the sequence in the multi-head self-attention layer of each ViT block.
- ADAPTER-k [8]: freezes all the backbone parameters while adding a down projection, a ReLU [7] non-linearity, and an up projection layer sequentially in the feed-forward network (FFN) of each visual Transformer block. We follow the training details of [16] to achieve better performance.
- LORA-k [9]: freezes all the backbone parameters while adding a concurrent branch including two low-rank matrices to the weight matrices in the multi-head self-attention layers to approximate efficiently updating them. The low-rank matrices can be merged into the backbone weights after fine-tuning. We follow the training details of [16] to achieve better performance.
- ADAPTFORMER [3]: freezes all the backbone parameters while adding a concurrent branch including a down projection, a ReLU [1] non-linearity, an up projection layer, and a pre-defined scaling factor to the FFN layer of each ViT block.
- NOAH [16]: searches for an optimal configuration with a once-for-all [2] network that includes trainable prompts, adapter modules, and LoRA modules, which requires a longer training schedule than the other VPET methods.

†Corresponding author. E-mail: bohan.zhuang@gmail.com

B. More Parameter Sensitivity Patterns

We show more parameter sensitivity patterns for ViT-B/16 with various pre-training strategies (i.e., MAE [6] and MoCo V3 [4]) and datasets sampled from FGVC benchmark [10]. We visualize the proportions of the sensitive parameters under 0.4M trainable parameter budget. Visualizations of sampled VTAB-1k datasets with MAE and MoCo V3 pre-trained ViT-B/16 are shown in Figures A, B, C. Visualizations of sampled FGVC datasets with supervised pre-trained ViT-B/16 are shown in Figure D. We find our observations in the main paper are general: the proportions of the sensitive parameter exhibit: 1) dataset-specific varying patterns in terms of network depth; and 2) dataset-agnostic similar patterns in terms of operations. We empirically find that the self-supervised pre-trained backbones have higher sensitivity variances than the supervised pre-trained one across the 19 downstream tasks. In particular, the variance of ViT-B/16 pre-trained with MAE [6] is twice as large as that of the supervised pre-trained ViT-B/16. We speculate that our SPT variants can better handle the large variances for self-supervised pre-trained backbones (Table 2 of the main paper) by identifying task-specific positions to introduce the trainable parameters.

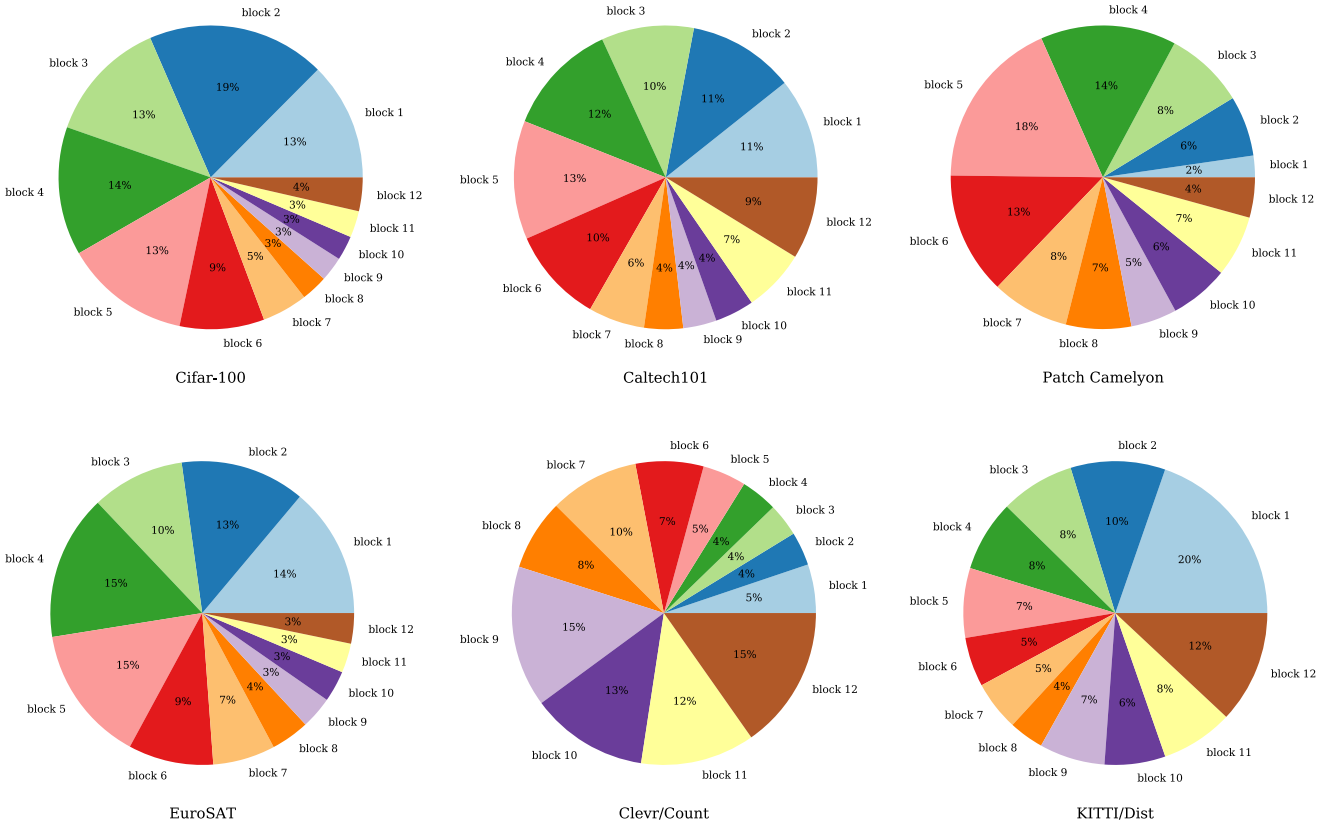

Figure A: The distribution of sensitive parameters by blocks under 0.4M trainable parameter budget with supervised pre-trained ViT-B/16 backbone. We sample six tasks from VTAB-1k [15].

C. Dataset Samples for the Source and Target Domains

We visualize some sampled images from the source domain (ImageNet [12]) and the target domains (VTAB-1k [15]) in Figure F. We observe that the images from the Natural tasks of VTAB-1k are relatively more similar to the source domain compared to those from the Structured tasks of VTAB-1k, which aligns with our observation that Structured tasks have large domain gaps. As structured tuning improves the performance of Structured datasets (Section 4.3 of the main paper), we speculate that structured tuning facilitates mitigating such large domain gaps.

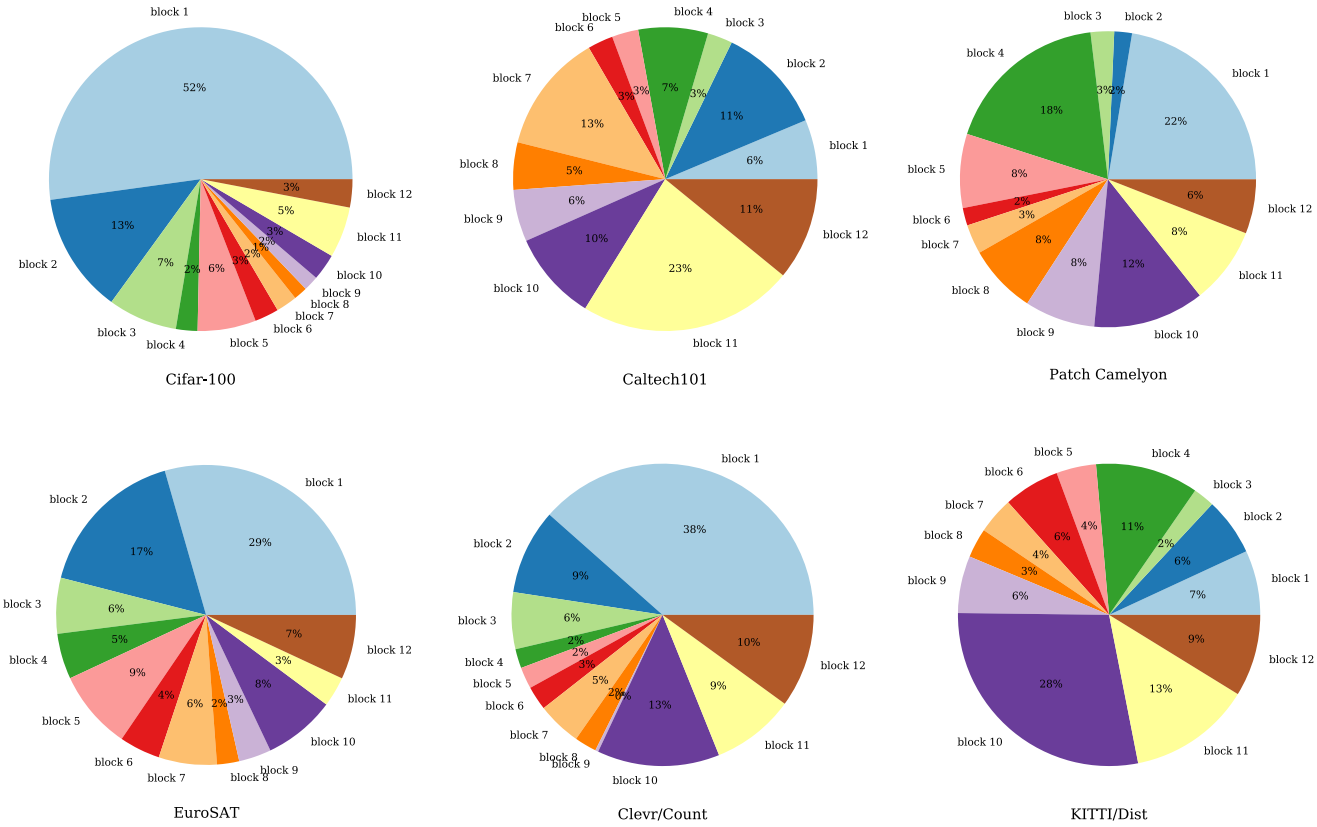

Figure B: The distribution of sensitive parameters by blocks under 0.4M trainable parameter budget with MAE [6] pre-trained ViT-B/16 backbone. We sample six tasks from VTAB-1k [15].

	Tuned / Total	CUB-200-2011	NABirds	Oxford Flowers	Stanford Dogs	Stanford Cars	Mean Acc.
FULL	100%	87.3	82.7	98.8	89.4	84.5	88.5
Addition-based methods							
MLP-3	1.50%	85.1	77.3	97.9	84.9	53.8	79.8
PROMPT-SHALLOW	0.31%	86.7	78.8	98.4	90.7	68.7	84.6
PROMPT-DEEP	0.98%	88.5	84.2	99.0	90.2	83.6	89.1
ADAPTER-8	0.39%	87.3	**84.3**	98.4	88.8	68.4	85.5
ADAPTER-32	0.95%	87.2	**84.3**	98.5	89.6	68.4	85.6
ADAPTFORMER	0.44%	84.7	75.2	97.9	84.7	83.1	85.1
SPT-ADAPTER	0.41%	**89.1**	83.3	**99.2**	90.5	85.6	89.5
SPT-ADAPTER	0.47%	**89.1**	83.3	**99.2**	**91.1**	**86.2**	**89.8**
Reparameterization-based methods							
LINEAR	0.12%	85.3	75.9	97.9	86.2	51.3	79.3
PARTIAL-1	8.38%	85.6	77.8	98.2	85.5	66.2	82.6
BIAS	0.13%	88.4	**84.2**	98.8	91.2	79.4	88.4
LORA-8	0.55%	84.9	79.0	98.1	88.1	79.8	86.0
LORA-16	0.90%	85.6	79.8	98.9	87.6	72.0	84.8
SPT-LORA	0.41%	**88.6**	82.8	99.4	**91.4**	84.5	89.3
SPT-LORA	0.60%	**88.6**	83.4	**99.5**	**91.4**	**87.3**	**90.1**

Table I: Per-task results on the FGVC benchmark from Table 1 of the main paper. "Tuned / Total" denotes the fraction of the trainable parameters. Top-1 accuracy (%) is reported. The best result is in **bold**, and the second-best result is underlined.

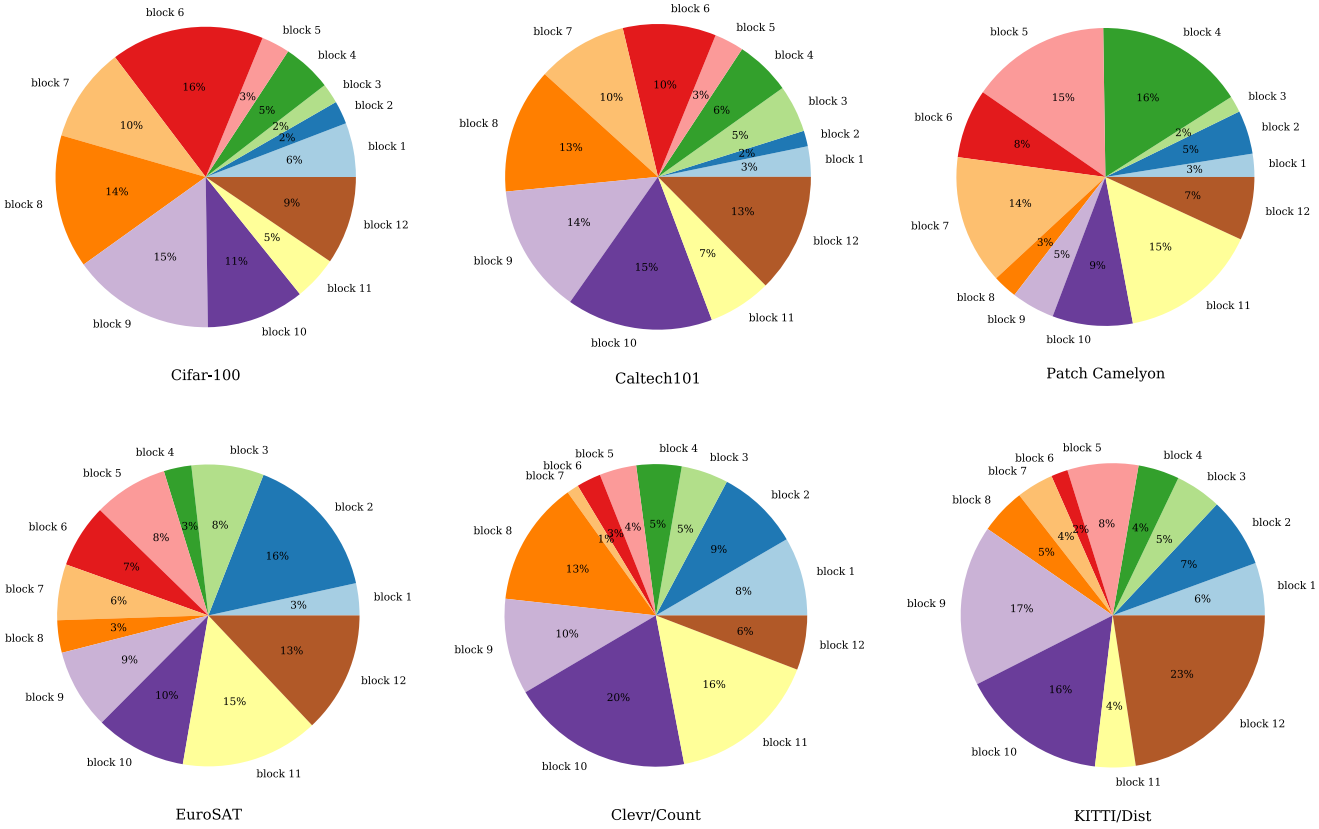

Figure C: The distribution of sensitive parameters by blocks under 0.4M trainable parameter budget for MoCo v3 [4] pre-trained ViT-B/16 backbone. We sample six tasks from VTAB-1k [15].

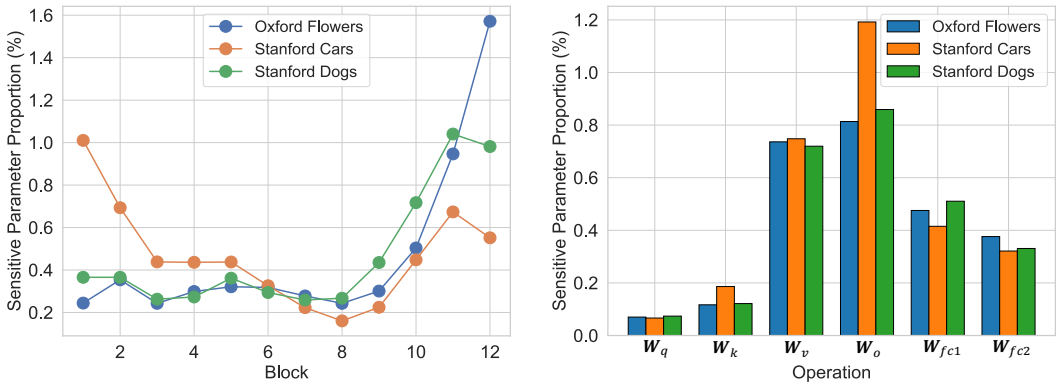

Figure D: Sensitivity patterns under 0.4M trainable parameters for Oxford Flowers [13], Stanford Cars [5], and Stanford Dogs [11]. We show the proportions of the sensitive parameters for the query $\boldsymbol{W}_q$, key $\boldsymbol{W}_k$, value $\boldsymbol{W}_v$, and $\boldsymbol{W}_o$ weight matrices in the multi-head self-attention layer and two weight matrices $\boldsymbol{W}_{fc1}$ and $\boldsymbol{W}_{fc2}$ in the feed-forward network.

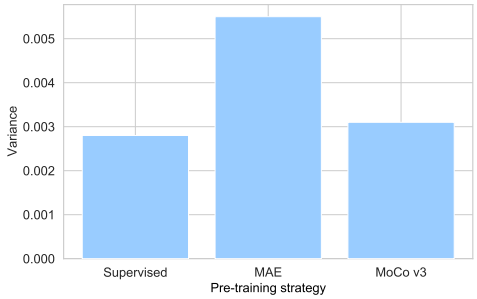

Figure E: Comparisons of sensitivity variances across backbones with different pre-training strategies on VTAB-1k.

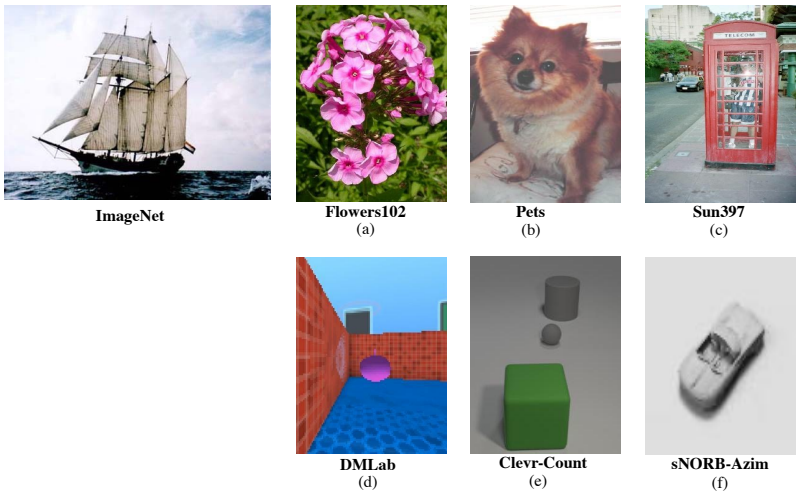

Figure F: Dataset samples from ImageNet [12] and VTAB-1k [15]. Samples from Natural tasks of VTAB-1k ((a), (b), and (c)) are relatively more similar to the source ImageNet samples compared to the ones from Structured tasks of VTAB-1k ((d), (e), and (f)).

		Natural								Specialized					Structured									
	Tuned / Total	**Cifar100**	**Caltech101**	**DTD**	**Flower102**	**Pets**	**SVHN**	**Sun397**	**Mean Acc.**	**Camelyon**	**EuroSAT**	**Resisc45**	**Retinopathy**	**Mean Acc.**	**Clevr-Count**	**Clevr-Dist**	**DMLab**	**KITTI-Dist**	**dSpr-Loc**	**dSpr-Ori**	**sNORB-Azim**	**sNORB-Ele**	**Mean Acc.**
FULL	100%	68.9	87.7	64.3	97.2	86.9	87.4	38.8	75.9	79.7	95.7	84.2	73.9	83.4	56.3	58.6	41.7	65.5	57.5	46.7	25.7	29.1	47.6
Addition-based methods																								
MLP-3	1.50%	63.8	84.7	62.3	97.4	84.7	32.5	49.2	67.8	77.0	88.0	70.2	56.1	72.8	47.8	32.8	32.3	58.1	12.9	21.2	15.2	24.8	30.6
PROMPT-SHALLOW	0.31%	77.7	86.9	62.6	97.5	87.3	74.5	51.2	76.8	78.2	92.0	75.6	72.9	79.7	50.5	58.6	40.5	67.1	68.7	36.1	20.2	34.1	47.0
PROMPT-DEEP	0.98%	78.8	90.8	65.8	98.0	88.3	78.1	49.6	78.5	81.8	96.1	83.4	68.4	82.4	68.5	60.0	46.5	72.8	73.6	47.9	32.9	37.8	55.0
ADAPTER-8	0.39%	69.2	90.1	68.0	98.8	89.9	82.8	54.3	79.0	84.0	94.9	81.9	75.5	84.1	80.9	65.3	48.6	78.3	74.8	48.5	29.9	41.6	58.5
ADAPTER-32	0.71%	68.7	92.2	69.8	98.9	90.3	84.2	53.0	79.6	83.2	95.4	83.2	74.3	84.0	81.9	63.9	48.7	80.6	76.2	47.6	30.8	36.4	58.3
NOAH	0.50%	69.6	92.7	70.2	99.1	90.4	86.1	53.7	80.2	84.4	95.4	83.9	75.8	84.9	82.8	68.9	49.9	81.7	81.8	48.3	32.8	44.2	61.3
SPT-ADAPTER	0.30%	72.9	93.2	72.5	99.3	91.4	84.6	55.2	81.3	85.3	96.0	84.3	75.5	85.3	82.2	68.0	49.3	80.0	82.4	51.9	31.7	41.2	60.8
SPT-ADAPTER	0.44%	72.9	93.2	72.5	99.3	91.4	88.8	55.8	82.0	86.2	96.1	85.5	75.5	85.8	83.0	68.0	51.9	81.2	82.4	51.9	31.7	41.2	61.4
Reparameterization-based methods																								
LINEAR	0.12%	63.4	85.0	63.2	97.0	86.3	36.6	51.0	68.9	78.5	87.5	68.6	74.0	77.2	34.3	30.6	33.2	55.4	12.5	20.0	9.6	19.2	26.8
PARTIAL-1	8.38%	66.8	85.9	62.5	97.3	85.5	37.6	50.6	69.4	78.6	89.8	72.5	73.3	78.5	41.5	34.3	33.9	61.0	31.3	32.8	16.3	22.4	34.2
BIAS	0.13%	72.8	87.0	59.2	97.5	85.3	59.9	51.4	73.3	78.7	91.6	72.9	69.8	78.3	61.5	55.6	32.4	55.9	66.6	40.0	15.7	25.1	44.1
LORA-8	0.55%	67.1	91.4	69.4	98.8	90.4	85.3	54.0	79.5	84.9	95.3	84.4	73.6	84.6	82.9	69.2	49.8	78.5	75.7	47.1	31.0	44.0	60.5
LORA-16	0.90%	68.1	91.4	69.8	99.0	90.5	86.4	53.1	79.8	85.1	95.8	84.7	74.2	84.9	83.0	66.9	50.4	81.4	80.2	46.6	32.2	41.1	60.2
SPT-LORA	0.31%	72.3	93.0	72.5	99.3	91.5	86.2	55.5	81.5	85.0	96.2	85.1	75.9	85.6	83.7	66.4	52.5	80.2	80.1	51.1	30.1	41.3	60.7
SPT-LORA	0.63%	73.5	93.3	72.5	99.3	91.5	87.9	55.5	81.9	85.7	96.2	85.9	75.9	85.9	84.4	67.6	52.5	82.0	81.0	51.1	30.2	41.3	61.3

Table II: Per-task results on the VTAB-1k benchmark from Table 1 of the main paper. "Tuned / Total" denotes the fraction of the trainable parameters. Top-1 accuracy (%) is reported.

References

[1] A. F. Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. 1

[2] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han. Once-for-all: Train one network and specialize it for efficient deployment. In *ICLR*, 2020. 1

[3] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *NeurIPS*, 2022. 1

[4] X. Chen, S. Xie, and K. He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9640–9649, 2021. 2, 4

[5] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng, and L. Fei-Fei. Fine-grained car detection for visual census estimation. In *AAAI*, 2017. 4

[6] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 2, 3

[7] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 1

[8] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, pages 2790–2799, 2019. 1

[9] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 1

[10] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim. Visual prompt tuning. In *ECCV*, 2022. 1, 2

[11] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPRW*, 2011. 4

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25, 2012. 1, 2, 5

[13] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, pages 722–729. IEEE, 2008. 4

[14] E. B. Zaken, Y. Goldberg, and S. Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL*, pages 1–9, 2022. 1

[15] X. Zhai, J. Puigcerver, A. Kolesnikov, P. Ruyssen, C. Riquelme, M. Lucic, J. Djolonga, A. S. Pinto, M. Neumann, A. Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. 1, 2, 3, 4, 5

[16] Y. Zhang, K. Zhou, and Z. Liu. Neural prompt search. *arXiv preprint arXiv:2206.04673*, 2022. 1