Appendix

1. Algorithm

Algorithm 1: k-medians on Standard Gaussian

Input: Bit-width b
Output: Sets $\mathcal{U}_1, ..., \mathcal{U}_{2^b}$ and codes $c_1, ..., c_{2^b}$

Initialize all c_i to 0;
Set $c_0 = -\infty, c_{2^b+1} = +\infty$;
while *not converged* **do**
  $\forall i, \mathcal{U}_i = (\frac{c_{i-1}+c_i}{2}, \frac{c_i+c_{i+1}}{2}]$;
  $\forall i, c_i =$ median of standard Gaussian within $\mathcal{U}_i$;
end
return $\mathcal{U}_1, ..., \mathcal{U}_{2^b}, c_1, ..., c_{2^b}$;

2. Derivation of Eq. (8)

If $i = k$,

$$\begin{aligned}
\frac{\partial \hat{w}_i}{\partial w_k} &= \frac{\partial(\hat{w_i}' \cdot \sigma + \mu)}{\partial w_i} \\
&= \sigma \cdot \frac{\partial \hat{w_i}'}{\partial w_i} + \hat{w_i}' \cdot \frac{\partial \sigma}{\partial w_i} + \frac{\partial \mu}{\partial w_i} \\
&= \sigma \cdot \frac{\partial w_i'}{\partial w_i} + \hat{w_i}' \cdot \frac{w_i'}{m} + \frac{1}{m} \\
&= \sigma \cdot \frac{\partial \frac{w_i - \mu}{\sigma}}{\partial w_i} + \hat{w_i}' \cdot \frac{w_i'}{m} + \frac{1}{m} \\
&= \sigma \cdot \frac{m - 1 - w_i'^2}{m\sigma} + \hat{w_i}' \cdot \frac{w_i'}{m} + \frac{1}{m} \\
&= 1 + \frac{w_i'(\hat{w_i}' - w_i')}{m}
\end{aligned}$$

Otherwise,

$$\begin{aligned}
\frac{\partial \hat{w}_i}{\partial w_k} &= \frac{\partial(\hat{w_i}' \cdot \sigma + \mu)}{\partial w_k} \\
&= \sigma \cdot \frac{\partial \hat{w_i}'}{\partial w_k} + \hat{w_i}' \cdot \frac{\partial \sigma}{\partial w_k} + \frac{\partial \mu}{\partial w_k} \\
&= \sigma \cdot \frac{\partial w_i'}{\partial w_k} + \hat{w_i}' \cdot \frac{w_k'}{m} + \frac{1}{m} \\
&= \sigma \cdot \frac{\partial \frac{w_i - \mu}{\sigma}}{\partial w_k} + \hat{w_i}' \cdot \frac{w_k'}{m} + \frac{1}{m} \\
&= \sigma \cdot \frac{-1 - w_i' w_k'}{m\sigma} + \hat{w_i}' \cdot \frac{w_k'}{m} + \frac{1}{m} \\
&= \frac{w_k'(\hat{w_i}' - w_i')}{m}
\end{aligned}$$

3. Supplementary Experiments

3.1. Performance on Full CIFAR100

In the VTAB-1K benchmark, each task only contains 1,000 training samples. We also conduct experiments on the full CIFAR100 [15] dataset, which has a larger 60,000-image training set. Following [17], we use a ViT-B/16 supervisedly pre-trained on ImageNet-21K with AugReg as backbone, and train the model for 100 epochs with batch size 128. We use $h = 8$ for AdaptFormer and $h = 32$ for Bi-AdaptFormer. All other settings are the same as in [17]. As shown in Table 1, compared to AdaptFormer, Bi-AdaptFormer brings 0.4% performance improvement and 8× more storage efficiency.

Method	Top-1 Acc.	Size (MB)
Full†	93.82	334.0
Linear†	88.70	0
BitFit†	93.39	0.39
VPT-Shallow†	90.38	0.59
VPT-Deep†	93.17	1.76
SSF†	**93.99**	0.78
AdaptFormer	93.55	0.56
Bi-AdaptFormer	93.95 (↑ 0.40)	**0.071**

Table 1: **Accuracy on full CIFAR100.** † denotes results reported in [17].

3.2. Semantic Segmentation

As for the dense prediction, we apply our method on Segmenter [28]. We use DeiT-B/16_{384} [29] pre-trained on ImageNet-1K as encoder. Since each segmentation task tunes an individual decoder upon the pre-trained encoder, we use a single FC layer as a decoder which is much more lightweight than FCN [20] and MaskTransformer [28]. We conduct experiments on Pascal-Context [22]. We evaluate three tuning paradigms: full fine-tuning, Adaptformer with $h = 8$, and Bi-AdaptFormer with $h = 32$. The models are trained for 50 epochs with batch size 64. As shown in Table 2, Bi-AdaptFormer still outperform AdaptFormer in terms of both performance and efficiency.

Method	mIoU (SS)	Size (MB)
Full	**52.61**	335.1
AdaptFormer	51.57	0.56
Bi-AdaptFormer	51.75 (↑ 0.18)	**0.071**

Table 2: **Semantic segmentation on Pascal-Context.** "Size" denotes the size of trainable parameters in encoders. We report mIoU of single-scale inference on validation set. Each method also has a decoder of 0.17MB.

3.3. Comparison with Other Quantization Methods

We compare our quantization method with existing binary neural networks – XNOR-Net [27] and IR-Net [26]. Similar to BI-ADAPTFORMER, we use the quantization strategy of XNOR-Net and IR-Net to quantize (*i.e.*, binarize) the weights of adapters to 1 bit, and keep the full-precision activation, called XNOR-ADAPTFORMER and IR-ADAPTFORMER, respectively. In experiments, we find that IR-ADAPTFORMER cannot be trained stably without Batch Normalization (BN) [10], so we add BN after each FC layer of the adapters. We set the hidden dimension $h = 8$ for ADAPTFORMER, and $h = 32$ for BI-ADAPTFORMER, XNOR-ADAPTFORMER, and IR-ADAPTFORMER.

As shown in Table 3, since IR-ADAPTFORMER uses additional BN and XNOR-ADAPTFORMER uses channel-wise scaling factors, their storage sizes are larger than that of BI-ADAPTFORMER. IR-ADAPTFORMER results in significant performance degradation compared to ADAPTFORMER. We conjunct this is because IR-Net is designed for traditional convolutional networks equiped with BN and is not suitable for plug-in adapters in modern large-size vision architecture. XNOR-ADAPTFORMER also underperforms ADAPTFORMER, demonstrating the necessity of tailoring quantization strategy for adapters.

Method	Avg. Acc.	Size (MB)
FULL	68.9	334.0
LINEAR	57.6	0
ADAPTFORMER	76.70	0.56
IR-ADAPTFORMER	72.13 (↓ 4.57)	0.14
XNOR-ADAPTFORMER	76.34 (↓ 0.36)	0.11
BI-ADAPTFORMER	**76.97** (↑ 0.27)	**0.071**

Table 3: **Average accuracy on VTAB-1K.**

4. Experimental Details

4.1. Datasets

See Table 5.

4.2. Pre-Trained Backbones

Model	Pre-Training Dataset	Size (M)	Pre-Trained Weights
ViT-B/16 [5]	ImageNet-21K	85.8	checkpoint
Swin-B [18]	ImageNet-21K	86.7	checkpoint
ConvNeXt-B [19]	ImageNet-21K	87.6	checkpoint
AugReg ViT-B/16 [5]	ImageNet-21K	85.8	checkpoint
DeiT-B/16_{384} [19]	ImageNet-1K	86.1	checkpoint

Table 4: **Pre-Trained backbones.**

4.3. Code Implementation

We use *PyTorch* and *timm* to implement all experiments on NVIDIA RTX 3090 GPUs.

4.4. Data Augmentation

4.4.1 VTAB-1K

Following [11], we just resize the images to 224×224.

4.4.2 Few-shot learning

Following [33], for training samples, we use color-jitter and RandAugmentation; for validation/test samples, we resize them to 256×256, crop them to 224×224 at the center, and then normalize them with ImageNet's mean and standard deviation.

4.4.3 Full CIFAR100

Following [17], we use a strong augmentation in the fine-tuning setting of [5]. Please refer to the official code of [5].

4.4.4 Semantic Segmentation

We completely follow the setting used in [28], which does mean substraction, random resizing, random left-right flipping, and randomly crops large images and pad small images to 480×480.

4.5. Hyper-parameters

s of (BI-)ADAPTFORMER, (BI-)LORA, and FACT is searched from $\{0.01, 0.1, 1, 10, 100\}$. See Table 6 for other hyper-parameters. We basically follow the hyper-parameters used by [33].

	Dataset	# Classes	Train	Val	Test
VTAB-1K [32]					
Natural	CIFAR100 [15]	100	800/1,000	200	10,000
	Caltech101 [6]	102			6,084
	DTD [4]	47			1,880
	Oxford-Flowers102 [24]	102			6,149
	Oxford-Pets [25]	37			3,669
	SVHN [23]	10			26,032
	Sun397 [31]	397			21,750
Specialized	Patch Camelyon [30]	2	800/1,000	200	32,768
	EuroSAT [8]	10			5,400
	Resisc45 [3]	45			6,300
	Retinopathy [13]	5			42,670
Structured	Clevr/count [12]	8	800/1,000	200	15,000
	Clevr/distance [12]	6			15,000
	DMLab [1]	6			22,735
	KITTI-Dist [7]	4			711
	dSprites/location [9]	16			73,728
	dSprites/orientation [9]	16			73,728
	SmallNORB/azimuth [16]	18			12,150
	SmallNORB/elevation [16]	18			12,150
Few-shot learning					
	Food-101 [2]	101	1/2/4/8/16 per class	20,200	30,300
	Stanford Cars [14]	196		1,635	8,041
	Oxford-Flowers102 [24]	102		1,633	2,463
	FGVC-Aircraft [21]	100		3,333	3,333
	Oxford-Pets [25]	37		736	3,669
Supplementary experiments					
	CIFAR100 (Full) [15]	100	60,000	-	10,000
	Pascal-Context [22]	59	4,996	5,104	-

Table 5: **Statistics of used datasets.**

	optimizer	batch size	learning rate	weight decay	# epochs	lr decay	# warm-up epochs
VTAB-1K	AdamW	64	1e-3	1e-4	100	cosine	10
Few-shot learning	AdamW	64	5e-3	1e-4	100	cosine	10

Table 6: **Hyper-parameters.**

References

[1] Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. *arXiv preprint*, arXiv:1612.03801, 2016. 3

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Proceedings of ECCV*, 2014. 3

[3] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE*, 2017. 3

[4] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of CVPR*, 2014. 3

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of ICLR*, 2021. 2

[6] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In

Proceedings of CVPR workshops, 2004. 3

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 2013. 3

[8] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 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*, 2019. 3

[9] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *Proceedings of ICLR*, 2017. 3

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of ICML*, 2015. 2

[11] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Proceedings of ECCV*, 2022. 2

[12] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of CVPR*, 2017. 3

[13] Kaggle and EyePacs. Kaggle diabetic retinopathy detection. 2015. 3

[14] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of CVPR workshops*, 2013. 3

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 3

[16] Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of CVPR*, 2004. 3

[17] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. In *Proceedings of NeurIPS*, 2022. 1, 2

[18] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of ICCV*, 2021. 2

[19] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of CVPR*, 2022. 2

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of CVPR*, 2015. 1

[21] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint*, arXiv:1306.5151, 2013. 3

[22] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan L. Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of CVPR*, 2014. 1, 3

[23] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Proceedings of NIPS Workshops*, 2011. 3

[24] M-E Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *Proceedings of CVPR*, 2006. 3

[25] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of CVPR*, 2012. 3

[26] Haotong Qin, Ruihao Gong, Xianglong Liu, Mingzhu Shen, Ziran Wei, Fengwei Yu, and Jingkuan Song. Forward and backward information retention for accurate binary neural networks. In *Proceedings of CVPR*, 2020. 2

[27] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Proceedings of ECCV*, 2016. 2

[28] Robin Strudel, Ricardo Garcia Pinel, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of ICCV*, 2021. 1, 2

[29] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of ICML*, 2021. 1

[30] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. *arXiv preprint*, arXiv:1806.03962, 2018. 3

[31] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of CVPR*, 2010. 3

[32] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, André Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. The visual task adaptation benchmark. *arXiv preprint*, arXiv:1910.04867, 2019. 3

[33] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. Neural prompt search. *arXiv preprint*, arXiv:2206.04673, 2022. 2