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

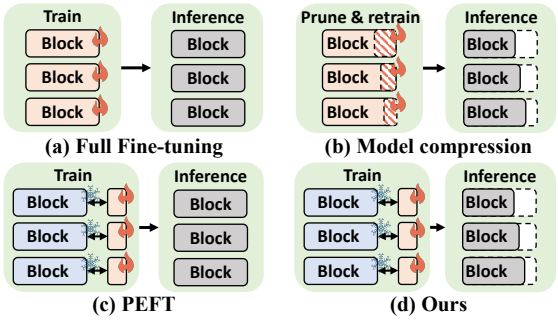
Abstract. Recently, the scale of transformers has grown rapidly, which introduces considerable challenges in terms of training overhead and inference efficiency in the scope of task adaptation. Existing works, namely Parameter-Efficient Fine-Tuning (PEFT) and model compression, have separately investigated the challenges. However, PEFT cannot guarantee the inference efficiency of the original backbone, especially for large-scale models. Model compression requires significant training costs for structure searching and re-training. Consequently, a simple combination of them cannot guarantee accomplishing both training efficiency and inference efficiency with minimal costs. In this paper, we propose a novel Parallel Yielding Re-Activation (PYRA) method for such a challenge of training-inference efficient task adaptation. PYRA first utilizes parallel yielding adaptive weights to comprehensively perceive the data distribution in downstream tasks. A re-activation strategy for token modulation is then applied for tokens to be merged, leading to calibrated token features. Extensive experiments demonstrate that PYRA outperforms all competing methods under both low compression rate and high compression rate, demonstrating its effectiveness and superiority in maintaining both training efficiency and inference efficiency for large-scale foundation models. Our code is available at https://github.com/THU-MIG/PYRA.

Keywords: Vision Transformer · Task Adaptation · Model Compression

1 Introduction

Vision transformers [16] have made a profound impact across various domains of computer vision, such as image classification [6,15,16,50,55,66], object detection [4,40,47,48,74], image segmentation [11,49,53,70,72], etc. In recent years, the

⋆ Corresponding Author



Method	(a)	(b)	(c)	(d)
Non-vast Data Scale	✓	✗	✓	✓
Training Efficient	✗	✗	✓	✓
Inference Efficient	✗	✓	✗	✓

Fig. 1: (a) Full fine-tuning trains all parameters on downstream tasks and utilizes the trained model for inference, thereby lacking efficiency in both training and inference stages. (b) Model compression employs pruning to enhance inference efficiency, but the pruned model necessitates extensive re-training on large-scale data. (c) PEFT freezes the model backbone and only fine-tunes a small amount of parameters, yet retains the inference complexity. (d) Our training-inference efficient task adaptation incorporates the advantages of all existing pipelines by training inference-efficient models with minimal tunable parameters.

scale of vision transformers has grown to be billion-parameters [29, 61, 66]. Consequently, adapting such models with vast scales into downstream tasks presents increasingly complex challenges, particularly in real-world deployment scenarios. Two critical concerns have been widely acknowledged as primary obstacles in implementing large-scale transformers for downstream applications [17, 23, 27, 46]: (1) the training overhead when fine-tuning on downstream tasks, and (2) the inference efficiency after model deployment.

Specifically, first, conventional fine-tuning methods, which necessitate adjusting all parameters of the model (*i.e.*, Full Fine-tuning), suffer from unaffordable consumption of GPU resources and training time given the extensive scales of the foundation models [16, 29, 66]. Researchers have delved into Parameter-Efficient Fine-Tuning (PEFT) [17, 19, 23, 24, 26, 36] algorithms, which generally freeze the pre-trained models and only tune extra small parameters, leading to great reduction of training time and storage overhead. The second issue pertains to inference efficiency, requiring the deployed model to promptly process the input data. The computational complexity of the models significantly influences achieving satisfactory inference throughput. Representative solutions for this matter encompass model compression methods, including model pruning [7, 10, 56, 64], knowledge distillation [1, 21, 52, 57, 58], model quantization [3, 13, 37], etc.

In the literature, these two intriguing topics are investigated separately. PEFT methods either identify a subset of tunable parameters in the backbone for fine-tuning [65] or introduce learnable parameters to the frozen backbone during fine-tuning [17, 23, 24, 26, 36]. While effectively reducing training costs, most of these methods inevitably escalate computational complexity, resulting in inefficient inference. Model compression methods frequently require signifi-

cant computational resources to identify optimal structures for pruning. After pruning, a comprehensive re-training process using a substantial amount of data is crucial to prevent significant performance degradation. Therefore, model compression methods are typically inefficient in training efficiency.

These observations naturally lead to the question: *can we achieve both training efficiency and inference efficiency simultaneously for downstream tasks?* We refer to this challenge as **Training-Inference Efficient Task Adaptation** (seeing Fig. 1). Exploring this issue can enable us to conveniently deploy the advanced large-scale foundation models in real-world downstream applications `with minimal costs`, which is appealing and essential for the widespread implementation of foundation models. A straightforward solution is combining PEFT and model compression. However, for efficient task adaptation, a heavy re-training stage is unaffordable. Consequently, simply combining PEFT and model compression can easily suffer from substantial performance drops. For instance, we could integrate LoRA [24], a notable PEFT method that introduces a small low-rank adapter, with ToMe [2], a parameter-free model compression technique for vision transformers[5]. As shown in Fig. 2, under lower compression rates (around 1.7×), performance on both backbones (ViT-L/16 and ViT-B/16) present slight drops (<1%) compared to directly conducting PEFT on the backbone, indicating that the ToMe+LoRA combination serves as a basic solution within lower compression rate range, yet performance improvements are also demanded. Under high compression rates (>3.0×), the performance quickly drops and is even inferior to directly fine-tuning the small-scale backbone with corresponding throughput. We term this phenomenon as *Adverse Compression*. Both phenomena indicate that directly combining existing works cannot effectively address the new challenges.

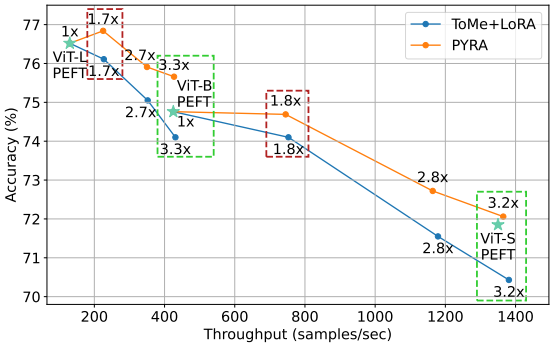


Fig. 2: Comparisons between simply combining ToMe [2] and LoRA [24] and our proposed PYRA. **Red boxes** represent the performance drop problem in low compression rates. **Green boxes** represent the adverse compression in high compression rates. See Sec. 4 for more results.

In this paper, we propose a novel **P**arallel **Y**ielding **R**e-**A**ctivation method (**PYRA**) designed for training-inference efficient task adaptation using vision transformers. Generally, the proposed PYRA follows the token merging paradigm [2, 8, 43] for inference efficiency. For effective task adaptation, we propose to mod-

[5] In Sec. 4, we show that ToMe+LoRA is a neat and strong solution.

ulate token features from both feature tokens and feature channels and parallelly yield weights for tokens to be merged by small modulation weight generators. These parallel yielding weights can comprehensively perceive the data distribution in downstream tasks. They are then applied to features through re-activation, resulting in adaptive token modulation. Thanks to such a token modulation strategy, PYRA can adaptively calibrate the learned feature distribution for downstream tasks with low computational complexity, ultimately leading to effective training and efficient inference.

We conduct extensive experiments to verify the effectiveness of our PYRA. We show that, under a low compression rate with $\sim$1.7$\times$ speedup, PYRA introduces negligible performance drops. Under a high compression rate with $>$3.0$\times$ speedup, PYRA eliminates the adverse compression gap. We emphasize that considering the scenario of high compression rates is practical, given the substantial size of current transformers. Concurrently, the specific model sizes required by downstream applications may not match any publicly available models, thus requiring the acquisition through high compression rates while keeping the performance comparable. In this regard, our approach represents an effective method for obtaining small models in the absence of pre-trained parameters for smaller-scale models.

Overall, we summarize our contribution as follows.

- We propose a novel challenge termed training-inference efficient task adaptation, in which the inference efficiency of large-scale transformers is escalated during parameter-efficient task adaptation.
- We propose PYRA for training-inference efficient task adaptation, which enhances the perception of feature distribution via token modulation. We generate parallel yielding decoupled weights to comprehensively perceive the feature distributions. We apply a re-activation strategy to modulate tokens to be merged for calibrated token features.
- Extensive experiments show that our PYRA outperforms all competing methods under both low compression rate and high compression rate. Further analysis shows that PYRA is effective across a series of different transformer backbones and model scales, well demonstrating the effectiveness and superiority of our method.

2 Related Works

Parameter-Efficient Fine-Tuning (PEFT) for Task Adaptation. Transferring large-scale transformers to downstream tasks has been a popular topic in computer vision [9, 17, 26, 28, 36, 60, 68, 69]. PEFT methods either locate a subset of parameters inside the model for fine-tuning [65], or inject human designed modules to the original model structure. Specifically, adapters [23, 28, 42, 45] are a type of MLP module with a bottleneck in the middle. Prompt-tuning [14, 18, 26, 32, 33, 39] insert learnable tokens to model input to generate task-specific outputs. SSF [36] tunes additional scaling and shifting parameters. LoRA [24], AdaptFormer [9], and Consolidator [17] add lightweight modules as

bypasses. These modules can be merged with the original backbone for no extra inference cost. In the scope of training-inference efficient task adaptation, PEFT methods retain model inference cost, resulting in extreme difficulty for model deployment. To solve the problem, we conduct adaptive token merging for PEFT. While achieving promising adaptation performance under both low and high compression rates, our method inherits the advantages of PEFT methods.

Model Compression. Model compression is often applied on large-scale models to acquire smaller-scale models with comparable performance. Mainstream approaches of model compression include model pruning [2, 8, 10, 54, 56, 64, 71], knowledge distillation [1, 21, 52] and model quantization [3, 13, 37, 38]. For transformer models, model pruning methods can be roughly grouped into two categories: channel pruning and token pruning. Channel pruning methods [5, 10, 64, 73] reduce the number of parameters, channels, heads, or blocks. Recently, token pruning has emerged as another mainstream approach. Several works have attempted to prune tokens for vision transformers (ViTs). Among these methods, token pooling [43] uses a slow k-means approach that does not work for an off-the-shelf model. ToMe [2] constructs bipartite graphs and merges token pairs with the most weighted connections. DiffRate [8] combines token pruning and token merging with searched optimal token reduction rates for each layer. Although achieving promising results, most existing model compression methods fail when combining with PEFT for training-inference efficient task adaptation since they usually involve a heavy training stage. Model pruning methods demand full re-training after pruning to restore performance. Knowledge distillation necessitates training a small-scale model from scratch, demanding large amount of data. As for model quantization, although no heavy training is demanded, post-training quantization [41, 63] achieves poor performance compared to quantization-aware training [25, 34] that demands full re-training on the quantized model. As a comparison, our proposed PYRA achieves promising performance on compressed models under the restrictions of PEFT, surpassing the baselines of simply combining model compression and PEFT.

3 Methodology

3.1 Preliminaries

ViT Model. In this paper, we mainly focus on the training-inference efficient task adaptation of ViT models [16, 20, 51]. A ViT model consists of L identical encoder blocks, each of which consists of a multi-head self-attention (MHSA) module and a feed-forward network (FFN). Formally, an input image $\mathbf{x}$ is reshaped and linear projected to N tokens $\mathbf{x} = [t_1, t_2, \cdots, t_N]$ in D dimensions. For simplicity, we omit the classification token ([CLS]) and the distillation token [51]. For encoder block l, we denote the input as $\mathbf{x}^l = [t_1^l, t_2^l, \cdots, t_{N^{l-1}}^l] \in \mathbb{R}^{N^{l-1} \times D}$ and the output as $\hat{\mathbf{x}}^l = [\hat{t}_1^l, \hat{t}_2^l, \cdots, \hat{t}_{N^l}^l] \in \mathbb{R}^{N^l \times D}$. For MHSA, the input tokens are first processed by three FC layers to generate Q,K, and V matrices, and the output is calculated by $\text{Softmax}(\frac{QK^T}{\sqrt{D}})V$ before being projected by another FC

layer. For FFN, the tokens are projected by two FC layers. Our method mainly focuses on the input tokens $\mathbf{x}^l$ before feeding them to the MHSA module.

LoRA. LoRA [24] is a widely employed PEFT method for task adaptation. Vision transformers consist of large dense parameter matrices. When adapting to a specific task, the updates to the matrices are in small subspaces and can be modeled with low-rank decompositions. LoRA trains only the decomposed matrices during fine-tuning. Specifically, for dense matrix $W_0 \in \mathbb{R}^{d\times k}$ and input x, the modified forward pass of updated W_0 is:

$$\hat{x} = W_0 x + BAx, \tag{1}$$

where $B \in \mathbb{R}^{d\times h}$ and $A \in \mathbb{R}^{h\times k}$. During inference, BA can be merged with W_0 for no extra computation overhead.

Token Merging. Token merging [2,8] is a parameter-free compression technique for vision transformers. It is orthogonal to the transformer structure and capable of flexibly changing the compression rate. Specifically, the input tokens $\mathbf{x}^l$ of the l-th ViT block are randomly separated before the MHSA:

$$G_1^l = [t_{i_1}^l, t_{i_2}^l, \cdots, t_{i_s}^l], \quad G_2^l = [t_{j_1}^l, t_{j_2}^l, \cdots, t_{j_s}^l], \quad 2s = N^{l-1}. \tag{2}$$

Then, for each $t_{i_\bullet}^l$ token, the most similar $t_{j_\bullet}^l$ token is matched to it via cosine similarity. From G_1^l, r tokens with the most similar connections to tokens in G_2^l are selected to form token pairs $(t_{m_k}^l, t_{n_k}^l)$, where $t_{m_k}^l \in G_1^l$, $t_{n_k}^l \in G_2^l$, and $k = 1, 2, \cdots, r$. Note that $\{m_k\}$ is a re-indexed subset of $\{i_\bullet\}$. Formally,

$$\{t_{m_k}^l\} = \arg\operatorname*{Top}_{\hat{t}_i^l} r(\max_j \frac{\hat{t}_i^l \cdot t_j^l}{\|\hat{t}_i^l\| \cdot \|t_j^l\|}), \quad t_{n_k}^l = \arg\max_{t_j^l}(\frac{t_{m_k}^l \cdot t_j^l}{\|t_{m_k}^l\| \cdot \|t_j^l\|}) \tag{3}$$

Previous works [2, 8, 43], generally merge tokens via average pooling, cutting down the computational cost by decreasing the token number by r in layer l.

We employ the above techniques as our baseline method, in which we attach token merging while fine-tuning LoRA for task adaptation. These methods are selected due to their advantages being highly compatible with training-inference efficient task adaptation. First, token merging is parameter-free and training-free, and does not change model structures, which retains the storage-efficient advantage of PEFT. Besides, we choose LoRA for its popularity, simplicity, and mergeability (not introducing extra FLOPs during inference). Experimental results show that ToMe+LoRA is a strong baseline (see Sec. 4).

3.2 PYRA: Parallel Yielding Re-Activation

Conventional fine-tuning methods excel at guiding the model to dynamically align with the target data distribution in downstream tasks by extensively adjusting parameters. However, in the context of training-inference efficient task adaptation, only a fraction of parameters can be fine-tuned, posing significant challenges in accurately capturing the nuances of data distribution. While reducing model complexity through token merging shows promise in enhancing

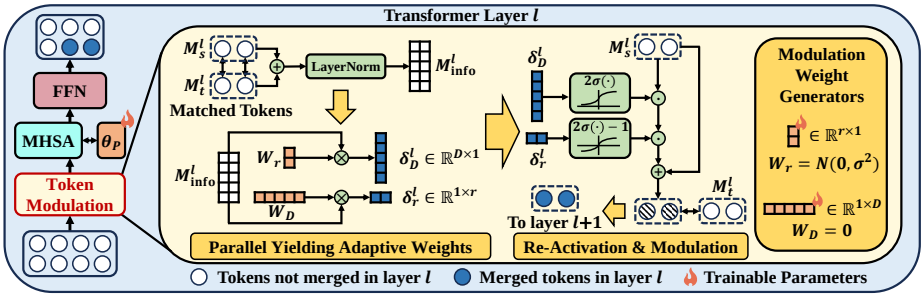


Fig. 3: The pipeline of our PYRA. PYRA conducts token modulation before the MHSA module in each transformer block. Inside PYRA, a pair of learnable modulation weight generators are leveraged to generate adaptive modulation weights parallelly. After that, generated weights modulate tokens through re-activation. The generators in PYRA can be trained along with the LoRA module θ_p in an end-to-end manner.

inference efficiency, it introduces the risk of information loss during layer-wise processing in vision transformers. This loss is difficult to rectify due to the constrained understanding of data distribution in the efficient task adaptation scenario. Therefore, a straightforward combination of token merging with PEFT algorithms for achieving training-inference efficient task adaptation may not yield optimal results, as depicted in Fig. 2.

Here, we propose **P**arallel **Y**ielding **R**e-**A**ctivation (**PYRA**) to adaptively modulate token features to enhance the perception of data distribution during token merging. Specifically, inside PYRA, the weights for adaptive merging are first yielded in a parallel manner through a pair of lightweight learnable vectors in each ViT block. These generated weights are then applied to modulate tokens to be merged through re-activation. As a result, PYRA enables adaptive calibration of the learned feature distribution with low computational complexity.

Parallel Yielding Adaptive Weights. We aim to optimize the merging process of each chosen token pair. Inspired by feature modulation [44, 59], we propose to modulate token features before merging. Formally, for encoder block l with r pairs of tokens in D dimensions to be merged, we group the $t_{m_k}^l$ and $t_{n_k}^l$ tokens as token matrices $M_s^l = [t_{m_1}^l, \cdots, t_{m_r}^l]$ and $M_t^l = [t_{n_1}^l, \cdots, t_{n_r}^l]$, where $M_s^l, M_t^l \in \mathbb{R}^{D\times r}$. We learn a modulation matrix $W^l \in \mathbb{R}^{D\times r}$ that adaptively modulates tokens at the granularity of each channel. We emphasize that directly learning W^l is redundant [24] and cannot adaptively satisfy the conditions with different images and token pairs. Therefore, we further specify the goal as learning decoupled weights $\delta_D^l = f_D(M_s^l, M_t^l) \in \mathbb{R}^{D\times 1}$ for feature channels and $\delta_r^l = f_r(M_s^l, M_t^l) \in \mathbb{R}^{1\times r}$ for feature tokens, where $W^l = \delta_D^l \delta_r^l$, separately. We propose to generate δ_D^l and δ_r^l in a parallel yielding manner.

Specifically, for layer l with r tokens to be merged, we create two learnable vectors as the modulation weight generator inside the transformer block: $W_r^l \in \mathbb{R}^{r\times 1}$ and $W_D^l \in \mathbb{R}^{1\times D}$. To guarantee that W_r^l and W_D^l digest the token features from both tokens in a token pair, we first normalize the sum of token pairs to

construct the token information matrix:

$$M_{\text{info}}^{l} = \text{LayerNorm}(M_s^l + M_t^l) \in \mathbb{R}^{D \times r}. \tag{4}$$

We normalize the distribution of M_{info}^{l} tokens by leveraging the LayerNorm$(\cdot)$ operation to enable smoother gradients when training W_r^l and W_D^l. With the token information matrix, we then yield the adaptive weights δ_D^l and δ_r^l parallelly:

$$\begin{aligned} \delta_D^l &= M_{\text{info}}^{l} W_r^l \in \mathbb{R}^{D \times 1} \\ \delta_r^l &= W_D^l M_{\text{info}}^{l} \in \mathbb{R}^{1 \times r}. \end{aligned} \tag{5}$$

Re-Activation for Token Modulation. Simply generating δ_D^l and δ_r^l via matrix multiplication still faces several possible issues. First, no measures have taken to ensure that δ_D^l and δ_r^l stay in a normal range. Second, decoupling weights to feature tokens and feature channels results in a low-rank modulation weight matrix W^l, which exhibits limited expressive capacity and thus could be unable to optimally modulate tokens in complicated data distributions. To cope with these issues, we conduct token modulation in a re-activation strategy. Specifically, we first broadcast δ_D^l to $\hat{\delta}_D^l \in \mathbb{R}^{D \times r}$ and conduct sigmoid activation $\sigma(\cdot)$ on it, and then modulate M_s^l for an intermediate modulation result:

$$\hat{M}_s^l = 2\sigma(\hat{\delta}_D^l) \odot M_s^l. \tag{6}$$

where $\odot$ is Hadamard product. $\hat{M}_s^l$ is then modulated with sigmoid-activated broadcast weight $\hat{\delta}_r^l \in \mathbb{R}^{D \times r}$ again to acquire the modulated tokens:

$$M_s^l \leftarrow M_s^l + (2\sigma(\hat{\delta}_r^l) - 1) \odot \hat{M}_s^l, \tag{7}$$

Note that we use the original tokens M_s^l to create a residual connection that preserves the gradient flow during training. The modulated M_s^l tokens are then merged with M_t^l with average pooling. We use a random Gaussian initialization for generator W_r^l and zero for generator W_D^l, so re-activation is equivalent to identity transformation at the beginning of training. Token modulation is conducted only on M_s^l to guarantee the parallelism in training and inference, as $t_{m_k}^l$ tokens are distinct while different $t_{n_k}^l$ might point to the same t_j^l token.

Discussion. Our PYRA effectively enhances the calibration of features for downstream tasks. Specifically, by utilizing parallel yielding adaptive weights, PYRA effectively decouples the modulation weight from the feature token and the feature channel, *i.e.*, δ_D and δ_r, enabling comprehensive perception of the feature distribution in downstream tasks. Furthermore, considering the challenge of capturing an accurate feature distribution for parameter-efficient training, such smooth re-activations by Eq. (6) and Eq. (7) can constrain the negative impact on weight values brought by limited perception of feature distributions, thus leading to better token modulation for improving feature representations in downstream tasks. As a result, the proposed PYRA can maintain discriminative information to an utmost degree during token merging, leading to improved performance while achieving complexity reduction.

Table 1: The complexity comparisons between conducting PEFT with and without PYRA. The FLOPs metric is obtained during inference.

Model	Metric	Total	PEFT w/o PYRA	(%)	PEFT w. PYRA	(%)
ViT-Base	# params	86M	0.29M	0.34%	0.30M	0.35%
	FLOPs	16.37G	16.37G	100%	8.15G	49.79%
ViT-Large	# params	303M	1.18M	0.39%	1.20M	0.40%
	FLOPs	57.37G	57.37G	100%	28.76G	50.13%

3.3 Complexity Analysis

We present the parameter and computation complexity for introducing PYRA to PEFT for task adaptation. In each ViT block, PYRA introduces $D + r$ training parameters and $4rD$ extra FLOPs. For a ViT model with L layers and R total merged tokens, in total, our PYRA introduces $LD+R$ extra training parameters and conducts $4RD$ extra FLOPs beside PEFT.

To better demonstrate the training-inference efficiency of our PYRA, we compare the complexity between attaching PYRA to task adaptation via PEFT (here we employ LoRA [24]) and plainly conducting PEFT without PYRA. As shown in Tab. 1, PYRA keeps the training efficient feature of PEFT by introducing only a tiny amount of training parameters, which however, results in a substantial efficiency boost for inference, *i.e.*, at around 50% for both ViT-B and ViT-L models. Results in Sec. 4 show that PYRA achieves comparable performance to the PEFT counterpart without PYRA. This well indicates that PYRA is an effective method for training-inference efficient task adaptation.

4 Experiments

4.1 Experimental Setting

Datasets. We conduct extensive experiments on task adaptation benchmarks to verify the effectiveness of PYRA within the challenge of training-inference efficient task adaptation. Specifically, we choose the VTAB-1k [67] benchmark for the evaluation. VTAB-1k is a challenging benchmark that consists of 19 different tasks from diverse domains: 1) natural images captured in the actual world; 2) specialized images from professional fields; and 3) structured synthesized images. Each task only contains 800 training samples and 200 validation samples.

Models. We choose two ViT backbones pre-trained on the ImageNet-21K [16], *i.e.*, ViT-L/16 and ViT-B/16, for comparison. Additionally, we also generalize our PYRA to different backbones and pre-train methods, including DeiT-B [51] and ViT backbones pre-trained by MAE [20].

Implementation Details. We choose LoRA [24] as the PEFT module for all methods due to its simplicity and mergeability. **For ease of explanation, we omit the "+LoRA" when comparing different methods.** We append LoRA only on the Q,K, and V projection matrices, and apply the training schedule for LoRA following [17, 69]. The generators are trained along with LoRA

Table 2: Results on the VTAB-1k [67] benchmark under low compression rate. **Bold** and underline denote the best and second-best accuracy within compression methods.

Method	# params	Throughput	Natural							Specialized				Structured								Average
			Cifar100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLAB	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	
Model: ViT-B/16 (Throughput: 425)																						
PEFT	0.34%	425	67.1	90.2	69.4	99.1	90.5	85.7	54.1	83.1	95.8	84.3	74.6	82.2	69.2	50.1	79.2	81.8	47.1	31.1	42.6	74.76
RaP	3.43%	654	25.9	68.4	53.3	64.0	57.4	71.3	21.5	75.8	87.9	59.3	73.6	43.1	53.8	26.3	60.5	73.5	25.5	16.7	27.9	55.57
SPViT	4.46%	567	41.6	75.4	61.1	83.2	66.2	56.1	28.3	79.3	94.2	73.3	73.6	70.6	61.5	42.4	67.8	75.4	**50.5**	28.9	31.3	64.16
DiffRate	0.35%	709	37.1	84.6	63.7	96.7	86.2	32.6	48.2	78.9	85.8	67.0	73.7	32.9	29.8	34.1	55.7	12.6	16.0	13.1	21.5	55.82
ToMe	0.34%	753	64.6	**90.4**	67.9	98.5	89.8	83.9	**53.2**	82.6	94.7	**83.5**	74.9	81.9	**69.8**	49.2	76.9	**81.9**	46.5	31.0	**43.1**	74.10
PYRA	0.35%	745	**67.5**	90.3	**69.3**	**98.9**	**90.0**	**84.6**	53.1	**83.3**	**95.7**	83.3	**75.2**	**82.6**	68.9	**50.8**	**80.0**	81.8	45.8	**32.2**	42.8	**74.69**
Model: ViT-L/16 (Throughput: 130)																						
PEFT	0.39%	130	77.1	91.4	73.4	99.5	91.3	89.6	57.6	85.9	96.1	87.3	76.1	83.1	63.0	50.7	82.1	81.7	53.5	32.2	36.6	76.52
RaP	1.95%	196	43.2	87.9	62.6	52.8	81.7	86.7	34.7	78.4	92.4	73.3	73.6	68.0	59.6	46.9	82.4	75.5	43.6	24.5	25.7	65.64
SPViT	2.47%	188	48.1	87.5	65.2	94.4	77.4	80.9	38.8	79.9	93.9	79.8	74.3	78.2	**65.8**	47.4	74.1	82.3	50.3	31.0	37.9	70.22
DiffRate	0.39%	221	50.9	86.8	70.3	97.8	88.3	39.0	52.3	80.2	87.2	72.2	74.2	32.6	32.3	36.5	57.4	22.8	26.6	15.2	23.4	59.53
ToMe	0.39%	227	76.1	91.1	72.3	99.2	**91.7**	89.2	56.4	86.4	95.1	86.6	75.1	82.4	61.9	50.9	81.4	81.6	**53.5**	33.4	36.8	76.11
PYRA	0.40%	225	**76.6**	**91.3**	**73.2**	**99.3**	91.5	**89.4**	**57.1**	**86.9**	**95.9**	**87.1**	**76.2**	**83.2**	63.2	**52.8**	**83.1**	**82.5**	52.6	**34.8**	**39.0**	**76.84**

modules. During inference, we merge the LoRA module to the backbone. All throughputs are measured during inference on a GeForce RTX 3090 GPU. More details can be found in the supplementary materials.

4.2 Performance comparison on Task Adaptation Benchmark

We verify the effectiveness of PYRA for training-inference efficient task adaptation on (1) low compression rate: comparing the performance of different methods under the same sparsity ratio (here we set sparsity ratio=50%); (2) high compression rate: comparing the performance between the competing methods and the smaller-scale model with similar throughput levels. We leverage ViT-L/16 and ViT-B/16 for these inspections. The competing methods include RaP [35], SPViT [30], DiffRate [8], and ToMe [2]. Comparisons to more baselines are in the supplementary materials. For RaP and SPViT, we first train the backbone with LoRA on downstream tasks, then we prune the models, and lastly we retrain the parameters attached by the pruning method and LoRA. DiffRate and ToMe can be employed with LoRA during fine-tuning, while DiffRate demands ImageNet-21K for searching the optimal compression schedule.

PYRA on low compression rate. Results on low compression rate are reported in Tab. 2. Overall, while achieving one of the best speedups on throughput, our PYRA is the best performed method compared to other competing methods using the lowest level of training parameters. Counting results on both backbones, our PYRA achieves the best or second-best performance on 37 of 38 dataset metrics. Compared to directly conducting PEFT on the backbone, while all competing methods cause worse results, our PYRA achieves comparable adaptation performance on ViT-B/16, and even outperforms ViT-L/16. The above results convincingly demonstrate that our PYRA successfully sets a new benchmark, *i.e.* reaching comparable performance as the uncompressed model,

Table 3: Results on the VTAB-1k [67] benchmark under high compression rate. **Bold** and underline denote the best and second-best accuracy within compression methods. ∗: As a comparison of similar throughputs, we compare ViT-B/16 with PEFT on ViT-S/16, and ViT-L/16 with PEFT on ViT-B/16.

Method	# params	Throughput	Natural							Specialized				Structured								Average
			Cifar100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLAB	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	
Model: ViT-B/16 (Throughput: 425)																						
PEFT*	0.34%	1350	57.7	88.2	70.1	98.7	88.7	85.7	44.9	81.4	94.7	84.6	73.6	81.6	64.1	48.1	80.0	72.9	38.4	22.9	37.7	71.85
RaP	0.86%	1029	24.3	40.1	34.5	41.8	40.5	21.7	11.4	75.8	86.5	35.1	73.8	49.6	49.7	28.1	39.4	13.8	15.4	12.4	26.9	42.60
SPViT	4.46%	944	23.7	67.9	51.9	69.9	53.2	19.6	13.1	71.9	81.3	67.9	74.7	53.5	61.9	39.5	57.4	45.0	34.5	11.1	23.2	52.49
DiffRate	0.35%	1308	23.2	73.0	55.7	87.9	66.7	27.2	29.3	78.1	77.8	53.1	73.6	29.7	28.6	31.7	52.6	11.5	17.2	11.3	20.3	49.29
ToMe	0.34%	1381	**54.2**	87.8	65.5	96.1	81.7	79.7	**45.2**	79.4	93.6	76.3	73.8	78.3	65.7	48.0	71.3	80.0	45.8	30.9	41.2	70.43
PYRA	0.35%	1365	54.0	**89.3**	**67.1**	**96.5**	**84.0**	**81.8**	44.6	**81.2**	**94.6**	**79.5**	**75.1**	**79.9**	**67.0**	**49.2**	**76.9**	**82.6**	**47.8**	**31.9**	**42.0**	**72.06**
Model: ViT-L/16 (Throughput: 130)																						
PEFT*	0.34%	425	67.1	90.2	69.4	99.1	90.5	85.7	54.1	83.1	95.8	84.3	74.6	82.2	69.2	50.1	79.2	81.8	47.1	31.1	42.6	74.76
RaP	0.65%	301	17.7	37.1	27.0	46.2	33.3	23.2	13.3	76.5	74.2	54.4	73.6	50.4	31.4	25.7	49.8	53.1	25.5	13.4	26.0	44.11
SPViT	2.47%	289	54.0	87.6	65.5	94.8	74.9	32.6	38.6	81.8	**95.3**	78.0	74.0	72.8	61.2	46.9	70.2	77.1	47.4	31.3	28.6	66.90
DiffRate	0.39%	416	47.4	73.5	54.1	84.3	60.2	19.6	22.2	50.0	64.6	42.8	18.2	31.5	31.9	31.1	37.3	22.0	17.7	14.8	21.4	40.48
ToMe	0.39%	431	71.0	90.9	70.4	98.3	88.5	87.2	**52.4**	82.9	94.5	83.1	75.0	80.7	61.1	48.9	76.9	80.8	53.0	32.1	35.2	74.10
PYRA	0.40%	427	**71.6**	**91.8**	**71.1**	**98.5**	**89.7**	**88.1**	52.2	**85.1**	**95.3**	**84.6**	**75.7**	**80.9**	**63.0**	**51.7**	**82.0**	**82.0**	**54.2**	**36.0**	**41.2**	**75.66**

on low compression rate for training-inference efficient task adaptation while accelerating the model to 1.75× speedup with only 0.4% training parameters.

PYRA on high compression rate. Results are reported in Tab. 3. With comparable throughputs to the smaller-scale model and minimal training parameters, PYRA outperforms all competing methods and achieves the best performance on 35 of 38 dataset metrics. Compared to the smaller-scale model with comparable throughput, PYRA successfully outperforms it. On the compressed ViT-L/16, PYRA even surpasses ViT-B/16 by 0.9%. This shows that PYRA, as the state-of-the-art, effectively bridges the adverse compression gap between compressed large-scale models and small-scale models, as mentioned in Sec. 1. Therefore, PYRA is an applicable alternative for acquiring a smaller-scale model (3.2× speedup) on downstream tasks through efficient training (0.4% training parameters) when no pre-trained small-scale model is available.

PYRA on self-supervised ViT backbone. We conduct the experiments on both compression rates on self-supervised ViT-L/16 (MAE) [20]. As shown in Tab. 4, PYRA significantly surpasses all competing methods. Under low compression rate, PYRA outperforms directly transferring the uncompressed backbone. Under high compression rate, PYRA also eliminates adverse compression. These results show that PYRA generalizes well for self-supervised visual models.

PYRA on different architectures. To further verify the generalizability of PYRA, we conduct the above experiments on the DeiT-B [51]. As shown in Tab. 5, PYRA achieves comparable performance to the uncompressed model under low compression rate and eliminates adverse compression under high compression rate, indicating its generalizability to other transformer architectures.

PYRA consistently yields better models of different throughputs. To further show the superiority of PYRA for different compression rates, we

Table 4: VTAB-1k [67] results on both compression rates for ViT-L (MAE) (Throughput: 130).

Model	Method	# params	Throughput	Average
ViT-L (MAE)	PEFT	0.39%	130	75.96
ViT-L (MAE)	RaP [35]	2.01%	182	67.55
	DiffRate [8]	0.39%	221	46.91
	ToMe [2]	0.39%	227	<u>74.97</u>
	PYRA	0.40%	225	**76.13**
ViT-B (MAE)	PEFT	0.34%	425	70.23
ViT-L (MAE)	RaP [35]	0.76%	298	52.91
	DiffRate [8]	0.39%	416	48.03
	ToMe [2]	0.39%	431	<u>68.20</u>
	PYRA	0.40%	427	**70.33**

Table 5: VTAB-1k [67] results on both compression rates for DeiT-B (Throughput: 431).

Model	Method	# params	Throughput	Average
DeiT-B	PEFT	0.34%	431	73.76
DeiT-B	RaP [35]	3.57%	695	62.34
	DiffRate [8]	0.35%	734	52.10
	ToMe [2]	0.34%	747	<u>72.83</u>
	PYRA	0.35%	740	**73.55**
DeiT-S	PEFT	0.34%	1332	70.01
DeiT-B	RaP [35]	1.09%	1187	57.70
	DiffRate [8]	0.35%	1314	44.51
	ToMe [2]	0.34%	1351	<u>68.68</u>
	PYRA	0.35%	1341	**70.13**

Table 6: Ablation study results for PYRA on ViT-B/16 under high compression rate. Here for # params we report only the parameters introduced by token modulation.

Method	W_r	W_D	Activation	# params	Natural	Specialized	Structured	Average
Baseline	×	×	×	0	72.87	80.78	57.64	70.43
Plain W_r	√	×	×	0.19K	72.90	81.07	57.66	70.54
$\sigma(\cdot)$ & W_r	√	×	√	0.19K	73.18	81.69	57.65	70.84
Plain W_D	×	√	×	8.45K	73.09	81.13	58.43	70.88
$\sigma(\cdot)$ & W_D	×	√	√	8.45K	73.31	<u>82.17</u>	58.44	71.31
Plain W_r & W_D	√	√	×	8.64K	<u>73.77</u>	81.37	<u>58.81</u>	<u>71.32</u>
PYRA	√	√	√	8.64K	**73.91**	**82.60**	**59.66**	**72.06**

compress the chosen backbones for a series of speedups and compare the results with the strongest baseline, ToMe [2], under similar values of throughputs. As shown in Figs. 2 and 4, PYRA consistently outperforms ToMe and flattens the accuracy-throughput curve. Therefore, our PYRA is applicable to acquire task-specific smaller-scale models of different throughputs consistently in the absence of pre-trained parameters for smaller-scale models.

4.3 Ablation Studies

We do controlled experiments to identify the effect of individual components in PYRA, *i.e.*, the generators W_r and W_D, and the sigmoid activation $\sigma(\cdot)$ applied on the generated modulation weights. Specifically, when using only a single generator, we omit Eq. (6) and replace $\hat{\delta}_r^l$ in Eq. (7) with the broadcast weight generated by the employed generator. When using the generators without sigmoid activation, we simply remove the σ in Eq. (6) and Eq. (7) while keeping other calculations intact. We choose the ImageNet-21K pre-trained ViT-B/16 to carry out the ablation studies under high compression rates as in Tab. 3.

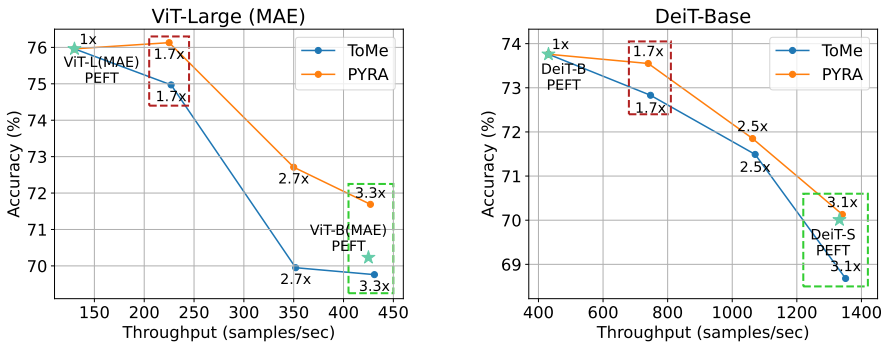


Fig. 4: Comparisons between PYRA and ToMe [2] under different compression rates for ViT-Large (MAE) and DeiT-Base. **Red boxes**: PYRA mitigates the performance drops under low compression rate. **Green boxes**: PYRA elinimates adverse compression under high compression rate.

Table 7: Adaptation performance comparison to $W_{r\times D}$ unadaptable to tokens. We report token modulation parameters.

Method	# params	Natural	Specialized	Structured	Average
Baseline	0	72.9	80.8	57.6	70.43
$W^l_{D\times r}$	70.66K	<u>73.8</u>	<u>81.5</u>	<u>59.2</u>	<u>71.49</u>
PYRA	8.64K	**73.9**	**82.6**	**59.7**	**72.06**

Table 8: Adaptation performance comparison to the common gated generator. We report token modulation parameters.

Method	# params	Natural	Specialized	Structured	Average
Baseline	0	<u>72.9</u>	80.8	57.6	70.43
Gated	73.73K	72.8	<u>81.5</u>	<u>58.9</u>	<u>71.06</u>
PYRA	8.64K	**73.9**	**82.6**	**59.7**	**72.06**

Results are shown in Tab. 6. Here the baseline method refers to conducting token merging [2, 8] while fine-tuning LoRA [24] for task adaptation. First, both W_r and W_D, no matter whether sigmoid activation is attached, lead to performance gains, and employing them simultaneously outperforms using them individually. This proves the superiority of our parallel yielding strategy. Second, compared to the corresponding counterpart without activations, employing sigmoid activation always leads to significant improvements, indicating the effectiveness of the re-activation. Overall, our PYRA yields the best adaptation performance, showing that our strategies are effective and complementary.

4.4 Further Analysis on Different Designs

We conduct further analysis on ViT-B/16 model pre-trained on ImageNet-21K. The baseline here refers to training LoRA with token merging [2, 8] attached. More analysis experiments can be found in the supplementary materials.

Impact of Making Modulation Weights Adaptive. In our PYRA, we train modulation weight generators W_D^l and W_r^l to conduct adaptive modulation on different merging tokens. To prove the necessity of making modulation weights adaptive, we compare PYRA with the approach of directly training final modulation weights $W_{D\times r}^l$ for each layer, and conduct token modulation as $M_s^l \leftarrow M_s^l + (2\sigma(W_{D\times r}^l) - 1) \odot M_s^l$. For fair comparisons, we inherit the sigmoid activation and the residual connection. As shown in Tab. 7, although with more training parameters, training $W_{D\times r}$ is still inferior to our PYRA. This indicates that PYRA is a more effective strategy to conduct adaptive token modulation.

Compare with the Common Gated Generator. We compare PYRA with setting the commonly-applied gated-style trainable module [12, 22, 31, 62] as modulation weight generator. Formally, for each ViT block, we insert a learnable two-layer MLP module ($\text{MLP}^l(\cdot)$) with input dimension D, hidden dimension $d \ll D$ to ensure parameter-efficient training and fast inference, and output dimension D. To generate modulation weights, we feed the information matrix M_{info}^l into the MLP, and modulate the merging tokens thereafter: $\delta^l = \text{MLP}(M_{\text{info}}^l) \in \mathbb{R}^{D\times r}, M_s^l \leftarrow M_s^l + (2\sigma(\delta^l - 1) \odot M_s^l$. We set $d = 4$ for all layers. As shown in Tab. 8, both the gated generator and our PYRA achieve performance gains, while our PYRA surpasses the gated generator by 1.0% with significantly fewer trainable parameters. This indicates that the decoupled weights generated by our W_D and W_r in PYRA are effective and parameter-efficient compared to the common gating strategy.

5 Conclusion

In this work, we defined and investigated a new challenge named training-inference efficient task adaptation, in which the inference efficiency of large-scale transformers is enhanced during parameter-efficient task adaptation. We propose a novel Parallel Yielding Re-Activation method (PYRA) to effectively cope with the challenge by modulating token features during token merging. Specifically, PYRA generates decoupled parallel yielding modulation weights, and conducts token modulation through re-activation. Extensive experiments show that PYRA introduces negligible performance drops under low compression rate, and bridges the gap of adverse compression between compressed transformers and small-scale models under high compression rate. In real-world applications, our PYRA is highly suitable to the scenario of transferring large-scale vision transformers to downstream tasks where no small-scale model is presented.

Limitations. We have not yet validated the effectiveness of PYRA in object detection and image segmentation. We plan to extend these tasks in the future. Meanwhile, it is also applicable to attach pruning methods after training the vision transformers with PYRA to eliminate additional parameters attached by PYRA. We leave that topic to future studies.

Acknowledgements

This work was supported by National Science and Technology Major 2022ZD0119401, National Natural Science Foundation of China (Nos. 62271281, 61925107, 62021002). It is also sponsored by CAAI-CANN Open Fund, developed on OpenI Community.

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. 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)
3. Cai, Y., Yao, Z., Dong, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Zeroq: A novel zero shot quantization framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13169–13178 (2020)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
5. Chavan, A., Shen, Z., Liu, Z., Liu, Z., Cheng, K.T., Xing, E.P.: Vision transformer slimming: Multi-dimension searching in continuous optimization space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4931–4941 (2022)
6. Chen, C.F.R., Fan, Q., Panda, R.: Crossvit: Cross-attention multi-scale vision transformer for image classification. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 357–366 (2021)
7. Chen, L., Zhao, H., Liu, T., Bai, S., Lin, J., Zhou, C., Chang, B.: An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. arXiv preprint arXiv:2403.06764 (2024)
8. 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)
9. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. Advances in Neural Information Processing Systems **35**, 16664–16678 (2022)
10. Chen, T., Cheng, Y., Gan, Z., Yuan, L., Zhang, L., Wang, Z.: Chasing sparsity in vision transformers: An end-to-end exploration. Advances in Neural Information Processing Systems **34**, 19974–19988 (2021)
11. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1290–1299 (2022)
12. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
13. Courbariaux, M., Bengio, Y., David, J.P.: Binaryconnect: Training deep neural networks with binary weights during propagations. Advances in neural information processing systems **28** (2015)

14. Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.M., Chen, W., et al.: Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. arXiv preprint arXiv:2203.06904 (2022)
15. Ding, Z., Wang, A., Chen, H., Zhang, Q., Liu, P., Bao, Y., Yan, W., Han, J.: Exploring structured semantic prior for multi label recognition with incomplete labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3398–3407 (2023)
16. 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)
17. Hao, T., Chen, H., Guo, Y., Ding, G.: Consolidator: Mergeable adapter with grouped connections for visual adaptation. arXiv preprint arXiv:2305.00603 (2023)
18. Hao, T., Ding, X., Feng, J., Yang, Y., Chen, H., Ding, G.: Quantized prompt for efficient generalization of vision-language models. arXiv preprint arXiv:2407.10704 (2024)
19. Hao, T., Lyu, M., Chen, H., Zhao, S., Han, J., Ding, G.: Re-parameterized low-rank prompt: Generalize a vision-language model within 0.5 k parameters. arXiv preprint arXiv:2312.10813 (2023)
20. 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)
21. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
22. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
23. 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)
24. 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)
25. Huang, X., Shen, Z., Cheng, K.T.: Variation-aware vision transformer quantization. arXiv preprint arXiv:2307.00331 (2023)
26. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European Conference on Computer Vision. pp. 709–727. Springer (2022)
27. Jiang, H., Wu, Q., Lin, C.Y., Yang, Y., Qiu, L.: Llmlingua: Compressing prompts for accelerated inference of large language models. arXiv preprint arXiv:2310.05736 (2023)
28. Jie, S., Deng, Z.H.: Convolutional bypasses are better vision transformer adapters. arXiv preprint arXiv:2207.07039 (2022)
29. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
30. Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Shen, X., Yuan, G., Ren, B., Tang, H., Qin, M., Wang, Y.: Spvit: Enabling faster vision transformers via latency-aware soft token pruning. 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 XI. Lecture

Notes in Computer Science, vol. 13671, pp. 620–640. Springer (2022). https://doi.org/10.1007/978-3-031-20083-0_37, https://doi.org/10.1007/978-3-031-20083-0_37
31. Kong, Z., Dong, P., Ma, X., Meng, X., Sun, M., Niu, W., Shen, X., Yuan, G., Ren, B., Qin, M., et al.: Spvit: Enabling faster vision transformers via soft token pruning. arXiv preprint arXiv:2112.13890 (2021)
32. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691 (2021)
33. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
34. Li, Y., Xu, S., Zhang, B., Cao, X., Gao, P., Guo, G.: Q-vit: Accurate and fully quantized low-bit vision transformer. Advances in Neural Information Processing Systems **35**, 34451–34463 (2022)
35. 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
36. Lian, D., Zhou, D., Feng, J., Wang, X.: Scaling & shifting your features: A new baseline for efficient model tuning. Advances in Neural Information Processing Systems **35**, 109–123 (2022)
37. Lin, X., Zhao, C., Pan, W.: Towards accurate binary convolutional neural network. Advances in neural information processing systems **30** (2017)
38. Lin, Z., Courbariaux, M., Memisevic, R., Bengio, Y.: Neural networks with few multiplications. arXiv preprint arXiv:1510.03009 (2015)
39. Liu, X., Ji, K., Fu, Y., Tam, W.L., Du, Z., Yang, Z., Tang, J.: P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. arXiv preprint arXiv:2110.07602 (2021)
40. 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)
41. Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., Gao, W.: Post-training quantization for vision transformer. Advances in Neural Information Processing Systems **34**, 28092–28103 (2021)
42. Lyu, M., Yang, Y., Hong, H., Chen, H., Jin, X., He, Y., Xue, H., Han, J., Ding, G.: One-dimensional adapter to rule them all: Concepts, diffusion models and erasing applications. arXiv preprint arXiv:2312.16145 (2023)
43. Marin, D., Chang, J.H.R., Ranjan, A., Prabhu, A., Rastegari, M., Tuzel, O.: Token pooling in vision transformers. arXiv preprint arXiv:2110.03860 (2021)
44. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Proceedings of the AAAI conference on artificial intelligence (2018)
45. Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., Gurevych, I.: Adapterfusion: Non-destructive task composition for transfer learning. arXiv preprint arXiv:2005.00247 (2020)
46. Song, Y., Zhou, Q., Li, X., Fan, D.P., Lu, X., Ma, L.: Ba-sam: Scalable bias-mode attention mask for segment anything model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3162–3173 (2024)

47. Song, Z., Jia, C., Yang, L., Wei, H., Liu, L.: Graphalign++: An accurate feature alignment by graph matching for multi-modal 3d object detection. IEEE Transactions on Circuits and Systems for Video Technology (2023)
48. Song, Z., Wei, H., Bai, L., Yang, L., Jia, C.: Graphalign: Enhancing accurate feature alignment by graph matching for multi-modal 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3358–3369 (2023)
49. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 7262–7272 (2021)
50. Tian, L., Ye, M., Zhou, L., He, Q.: Clip-guided black-box domain adaptation of image classification. Signal, Image and Video Processing **18**(5), 4637–4646 (2024)
51. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International conference on machine learning. pp. 10347–10357. PMLR (2021)
52. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1365–1374 (2019)
53. Wang, A., Chen, H., Lin, Z., Pu, H., Ding, G.: Repvit: Revisiting mobile cnn from vit perspective. arXiv preprint arXiv:2307.09283 (2023)
54. Wang, A., Chen, H., Lin, Z., Zhao, S., Han, J., Ding, G.: Cait: Triple-win compression towards high accuracy, fast inference, and favorable transferability for vits. arXiv preprint arXiv:2309.15755 (2023)
55. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 568–578 (2021)
56. Wang, Z., Luo, H., Wang, P., Ding, F., Wang, F., Li, H.: Vtc-lfc: Vision transformer compression with low-frequency components. Advances in Neural Information Processing Systems **35**, 13974–13988 (2022)
57. Wei, K., Du, R., Jin, L., Liu, J., Yin, J., Zhang, L., Liu, J., Liu, N., Zhang, J., Guo, Z.: Video event extraction with multi-view interaction knowledge distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 19224–19233 (2024)
58. Wei, K., Jin, L., Zhang, Z., Guo, Z., Li, X., Liu, Q., Feng, W.: More than syntaxes: Investigating semantics to zero-shot cross-lingual relation extraction and event argument role labelling. ACM Transactions on Asian and Low-Resource Language Information Processing **23**(5), 1–21 (2024)
59. Wu, Y., He, K.: Group normalization. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
60. Xiong, Y., Chen, H., Lin, Z., Zhao, S., Ding, G.: Confidence-based visual dispersal for few-shot unsupervised domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11621–11631 (2023)
61. Xiong, Y., Chen, X., Ye, X., Chen, H., Lin, Z., Lian, H., Niu, J., Ding, G.: Temporal scaling law for large language models. arXiv preprint arXiv:2404.17785 (2024)
62. Xu, L., Xie, H., Qin, S.Z.J., Tao, X., Wang, F.L.: Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. arXiv preprint arXiv:2312.12148 (2023)
63. Yao, Z., Yazdani Aminabadi, R., Zhang, M., Wu, X., Li, C., He, Y.: Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. Advances in Neural Information Processing Systems **35**, 27168–27183 (2022)

64. Yu, S., Chen, T., Shen, J., Yuan, H., Tan, J., Yang, S., Liu, J., Wang, Z.: Unified visual transformer compression. arXiv preprint arXiv:2203.08243 (2022)
65. 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)
66. Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L.: Scaling vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12104–12113 (2022)
67. 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)
68. Zhang, Y., Qiu, L., Zhu, Y., Wen, L., Luo, X.: A new childhood pneumonia diagnosis method based on fine-grained convolutional neural network. Computer Modeling in Engineering & Sciences **133**, 873–894 (2022)
69. Zhang, Y., Zhou, K., Liu, Z.: Neural prompt search. arXiv preprint arXiv:2206.04673 (2022)
70. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6881–6890 (2021)
71. Zhou, F., Zhou, Q., Yin, B., Zheng, H., Lu, X., Ma, L., Ling, H.: Rethinking impersonation and dodging attacks on face recognition systems. In: Proceedings of the 30th ACM International Conference on Multimedia (ACM MM) (2024)
72. Zhou, Q., Feng, Z., Gu, Q., Pang, J., Cheng, G., Lu, X., Shi, J., Ma, L.: Context-aware mixup for domain adaptive semantic segmentation. IEEE Transactions on Circuits and Systems for Video Technology **33**(2), 804–817 (2022)
73. Zhu, M., Tang, Y., Han, K.: Vision transformer pruning. arXiv preprint arXiv:2104.08500 (2021)
74. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)