Token Compensator: Altering Inference Cost of Vision Transformer without Re-Tuning

Shibo Jie[1], Yehui Tang[2], Jianyuan Guo[2], Zhi-Hong Deng[1*], Kai Han[2*], and Yunhe Wang[2*]

[1] State Key Laboratory of General Artificial Intelligence, School of Intelligence Science and Technology, Peking University
[2] Huawei Noah's Ark Lab

Abstract. Token compression expedites the training and inference of *Vision Transformers* (ViTs) by reducing the number of the redundant tokens, *e.g.*, pruning inattentive tokens or merging similar tokens. However, when applied to downstream tasks, these approaches suffer from significant performance drop when the compression degrees are mismatched between training and inference stages, which limits the application of token compression on off-the-shelf trained models. In this paper, we propose a model arithmetic framework to decouple the compression degrees between the two stages. In advance, we additionally perform a fast parameter-efficient self-distillation stage on the pre-trained models to obtain a small plugin, called **To**ken **Com**pensator (`ToCom`), which describes the gap between models across different compression degrees. During inference, `ToCom` can be directly inserted into any downstream off-the-shelf models with any mismatched training and inference compression degrees to acquire universal performance improvements without further training. Experiments on over 20 downstream tasks demonstrate the effectiveness of our framework. On CIFAR100, fine-grained visual classification, and VTAB-1k benchmark, `ToCom` can yield up to a maximum improvement of 2.3%, 1.5%, and 2.0% in the average performance of DeiT-B, respectively.

1 Introduction

Vision Transformers (ViTs) [8] have achieved remarkable success in various fields of computer vision, including image classification [38], object detection [5, 22], semantic segmentation [36], *etc*. However, with the rapid growth in the scale of ViTs, the increasing computational cost has become a pressing issue. Consequently, a large number of efforts are focusing on accelerating the training and inference of ViTs [2, 7, 26–28, 45]. The characteristic of ViTs lies in their capacity to accommodate a variable number of input tokens. Thus, beyond the conventional techniques widely utilized in convolutional neural networks such as model pruning, quantization, and distillation, recent researches suggest the acceleration of ViTs through token compression, such as pruning inattentive tokens [9, 24, 42, 43] or merging similar tokens [1, 30, 40].

*Corresponding Author.

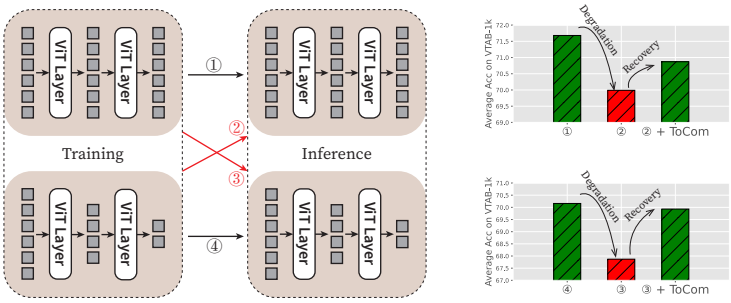

Fig. 1: ***Left:*** Previous token compression methods focus on scenario when training and inference compression degrees are consistent (① and ④), but not adequately address the performance of models when these degrees differ (② and ③). ***Right:*** Performance of token compression significantly degrades when compression degrees in training and inference are not equal. After applying our `ToCom` without training, the performance is recovered.

Token compression techniques offer distinct advantages. In comparison to techniques like pruning and distillation, some token compression approaches (*e.g.*, ToMe [1]) can be applied in a zero-shot manner to off-the-shelf models or utilized for accelerating training; unlike quantization, token compression methods do not necessitate support for low-precision operators. Furthermore, token compression methods operate orthogonally to the aforementioned other techniques, rendering them widely applicable in ViTs.

However, we observe the following drawbacks in token compression when applied to downstream tasks. Firstly, although some token compression techniques can be applied to off-the-shelf models, they often lead to significant performance degradation. Secondly, even if token compression is only applied during training to expedite the process, and tokens are not compressed during inference, the model's performance still falls below models trained without token compression. In summary, *when there is inconsistency in the token compression degrees between training and inference stages, the performance of models is suboptimal*, as illustrated in Fig. 1. This limitation restricts its application across various scenarios. For example, if we aim to dynamically adjust the computation costs of deployed models based on server load, we would need to train a model for each candidate of computation costs and each downstream task to achieve optimal performance, resulting in significant training and storage overheads. Additionally, if we intend to accelerate training with minimal loss in model inference performance, token compression in training-time only may lead to obvious performance drop.

In this paper, we point out that models fine-tuned under different token compression degrees exhibit a certain gap at the parameter level, causing performance degradation when altering compression degrees during inference. Also, we observe that this gap can be transferred across different downstream datasets. Motivated by this, we propose **To**ken **Com**pensator (`ToCom`), a pre-trained plu-

gin designed to decouple the token compression degrees between training and inference, to address the aforementioned challenges. ToCom is a parameter-efficient module that only contains a negligible number of parameters, which describes the gap between models with different compression degrees. To obtain ToCom, we train it on pre-training datasets with a fast self-distillation process among different compression degrees. To elaborate, both the teacher model and the student model are the same frozen pre-trained model, with the student model incorporating ToCom. Different compression degrees are randomly assigned to the teacher and student models in each step, while ToCom learns the gap between them through distillation. Moreover, we allocate different subsets of ToCom parameters for different compression degree pairs, enabling ToCom to adapt to various compression degree pairs through a single training process.

During inference, we directly integrate ToCom into off-the-shelf models fine-tuned on downstream tasks without any further training. By selecting the subsets of ToCom parameters, the fine-tuned model can be directly applied to various token compression degrees and achieve on-par performance comparable to the case when training and inference compression degrees are consistent. Importantly, ToCom only needs to be pre-trained once and can be applied to models fine-tuned on arbitrary downstream datasets with arbitrary token compression degrees, thereby enable any single off-the-shelf model to handle dynamic latency constraints without parameter modification.

We conduct experiments on over 20 datasets across various settings of compression degree. Experimental results demonstrate that ToCom, as a plug-and-play module, can effectively decouple the compression degree between training and inference for token compression. For example, on VTAB-1k benchmark, ToCom can yield up to a maximum improvement over ToMe of 2.0% in the average performance of DeiT-B, as shown in Fig. 1. ToCom can also be applied to models with different scales or models pre-trained with different objects, or utilized to enhance various token compression methods including token merging and token pruning.

2 Related Work

Token Compression Token compression aims to eliminate redundancy by reducing the number of tokens, thereby expediting ViTs. It primarily encompasses two directions: token pruning and token merging. Token pruning involves assessing token importance through defined metrics and removing unimportant tokens accordingly. PS-ViT [37] proposes a top-down paradigm to estimate the importance of each token. DynamicViT [32] fine-tunes lightweight subnetwork to evaluate tokens. EViT [24] and Evo-ViT [42] use the attention score of [CLS] tokens as a metric of attentiveness, and optionally pools all the pruned tokens into one. A-ViT [43] and ATS [9] also adjust the pruning rate based on the complexity of the input image, but their dynamic token length is not well-supported by current computation framework. SuperViT [25] trains a single ViT to support various token keeping rates.

On the other hand, token merging reduces token quantity by merging similar tokens. Token Pooling [30] uses k-means for token clustering. ToMe [1] proposes a Bipartite Soft Matching (BSM) algorithm to gradually merge similar tokens. TPS [40] achieves token merging by performing pruning first, and fusing the reserved token and pruned tokens. CrossGET [35] propose Complete-Graph Soft Matching and Cross-Guided Matching for token merging. There are also methods that combine pruning and merging together, such as DiffRate [3] and PPT [41].

Among the aforementioned methods, some methods are parameter-free, such as ToMe, ATS, CrossGET, and EViT, and thus can be directly used on off-the-shelf models in inference stage. Moreover, since these methods also reduce the tokens during training, they can be leveraged to accelerate the training stage. However, current research has not thoroughly explored the potential of token compression methods in these aspects.

Parameter-Efficient Tuning *Parameter-Efficient Tuning* (PET) aims to fine-tune pre-trained vision backbones for downstream tasks by adjusting only a small subset of parameters with backbone frozen, including prompt-based methods which append trainable tokens to the sequential inputs of transformers as prompts [11, 16, 44]; adapter-based methods which incorporate small adapters into the pre-trained model [4, 12, 14, 17–19, 33]; tuning bias parameters only [46]; altering intermediate features using affine transformation [23]; and matching feature changes in fine-tuning with a small side-network [48].

Previous PET methods focus on adaptation, in other words, using small PET modules to characterize the gap between the pre-trained and fine-tuned models. In contrast, we take a different perspective in this paper, using the PET modules to describe the gap between models with different compression degrees, serving as a universal compensator.

Model Arithmetic Model arithmetic involves merging, enhancing, or removing specific capabilities between different models through model-level addition or subtraction. [15, 49] show that arithmetic of models can achieve distribution generalization, multi-task learning, unlearning, and domain transfer. Some works on diffusion model have also utilized model arithmetic to reduce iterative steps [29] or combine the style and object for generating [34]. However, the application of model arithmetic on token compression has not been explored yet.

3 Delve into Token Compression

3.1 Impact of Compression Degrees

First, we formulate the token compression method on ViTs. A single layer of ViTs consists of two blocks, namely Multi-Head Self-Attention (MHSA) and Multi-Layer Perceptron (MLP). The layer can be formalized as

$$\widetilde{\mathbf{X}}^l = \mathbf{X}^l + \mathrm{MHSA}(\mathrm{LN}(\mathbf{X}^l)), \quad \mathbf{X}^{l+1} = \widetilde{\mathbf{X}}^l + \mathrm{MLP}(\mathrm{LN}(\widetilde{\mathbf{X}}^l)), \tag{1}$$

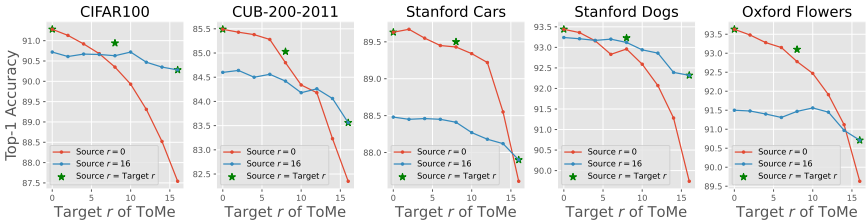

Fig. 2: Performance of ToMe on CIFAR100 and FGVC datasets. We use DeiT-B as pre-trained backbone. We report performance when source $r \in \{0, 16, \text{target } r\}$

.

Table 1: Results of model gaps transfer. We use CIFAR100 as $\mathcal{D}_\mathcal{A}$ and FGVC tasks as $\mathcal{D}_\mathcal{B}$. The results are evaluated with $r = 16$.

Dataset $\mathcal{D}_\mathcal{B}$	$\mathcal{M}_0^{\mathcal{D}_\mathcal{B}}$	$\mathcal{M}_{16}^{\mathcal{D}_\mathcal{A}} - \mathcal{M}_0^{\mathcal{D}_\mathcal{A}} + \mathcal{M}_0^{\mathcal{D}_\mathcal{B}}$
CUB-200-2011	82.4	**83.2**
Stanford Cars	87.6	**87.9**
Stanford Dogs	89.5	**90.3**
Oxford Flowers	89.6	**91.2**

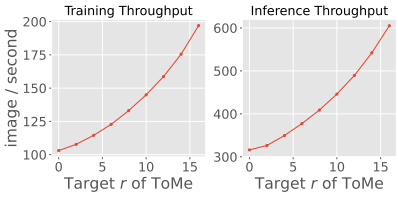

Fig. 3: Training and inference throughput of DeiT-B with different r of ToMe. Batch size is 128 and 256 for training and inference, respectively.

in which $\mathbf{X}^l \in \mathbb{R}^{N \times d}$ is the input of the l-th layer with length N and dimension d, and LN denotes layer normalization.

In this paper, we mainly focus on a representative and state-of-the-art training-free token compression methods, ToMe [1], and then generalize to other methods. ToMe operates between MHSA and MLP blocks. It leverages the keys of patch tokens to evaluate their similarity, and merges r similar ones with Bipartite Soft Matching.

In ToMe, the number of merged tokens per layer are considered as hyper-parameters to adjust the throughput of the ViTs, which is usually determined based on inference requirements before training. The more tokens it merges, the faster the models in training and inference, as shown in Fig. 3. However, in real-world scenarios, the compression degrees during training (called **source degrees**) and inference (called **target degrees**) may not necessarily be equal. That is, an off-the-shelf model trained at one compression degree might be applied at different compression degrees without retraining. This scenario holds practical significance, such as when utilizing downloaded checkpoints without access to training data or resource for retraining, or dynamically adjusting the compression degree during inference based on server loads. Additionally, in cases where existing computational resources are limited, one may have to use a high compression degree to reduce memory and time overheads during training but restore to a lower compression degree during inference to ensure performance.

To investigate the performance of token compression methods when there is inconsistency between source and target degrees, experiments are conducted on five downstream datasets [10, 20, 21, 31, 39]. As illustrated in the Fig. 2, we fine-tune DeiT-B [38] with ToMe of $r = 0$ and 16, and report the performance during inference with $r = 0, 2, 4, ..., 16$. We observe that for a specific target degree, the model performs better when the source degree matches it. The greater the disparity between the source and target degrees, the more the performance degradation becomes.

However, since models trained at lower compression degrees have seen more tokens during training, implying they have encountered a broader range of information than models trained at higher compression degrees, the former ideally should outperform the latter on various target degrees. *This suggests the presence of a gap between models trained at different source degrees, making the transfer between different compression degrees less effective.*

3.2 Transfer across Tasks

For the gap between models with different source degrees, we pose a question: *Is this gap transferable across tasks?* More concretely, let $\mathcal{M}_m^{\mathcal{D}_\mathcal{A}}$ and $\mathcal{M}_n^{\mathcal{D}_\mathcal{A}}$ denote models trained with compression degree m and n on dataset $\mathcal{D}_\mathcal{A}$, and $\mathcal{M}_m^{\mathcal{D}_\mathcal{B}}$ and $\mathcal{M}_n^{\mathcal{D}_\mathcal{B}}$ denote models trained on dataset $\mathcal{D}_\mathcal{B}$. If the gap is transferable, we should have

$$\mathcal{M}_m^{\mathcal{D}_\mathcal{A}} - \mathcal{M}_n^{\mathcal{D}_\mathcal{A}} \approx \mathcal{M}_m^{\mathcal{D}_\mathcal{B}} - \mathcal{M}_n^{\mathcal{D}_\mathcal{B}}, \tag{2}$$

in which $+$ and $-$ are parameter-wise addition and subtraction, respectively. To validate this, we rewrite Eq. 2 as

$$\mathcal{M}_m^{\mathcal{D}_\mathcal{A}} - \mathcal{M}_n^{\mathcal{D}_\mathcal{A}} + \mathcal{M}_n^{\mathcal{D}_\mathcal{B}} \approx \mathcal{M}_m^{\mathcal{D}_\mathcal{B}}, \tag{3}$$

which means $(\mathcal{M}_m^{\mathcal{D}_\mathcal{A}} - \mathcal{M}_n^{\mathcal{D}_\mathcal{A}} + \mathcal{M}_n^{\mathcal{D}_\mathcal{B}})$ should perform better than $\mathcal{M}_n^{\mathcal{D}_\mathcal{B}}$ when evaluated on $\mathcal{D}_\mathcal{B}$ with target degree m. In other words, the gap $(\mathcal{M}_m^{\mathcal{D}_\mathcal{A}} - \mathcal{M}_n^{\mathcal{D}_\mathcal{A}})$ on $\mathcal{D}_\mathcal{A}$ can be transferred to $\mathcal{D}_\mathcal{B}$ and help to bridge the gap between $\mathcal{M}_m^{\mathcal{D}_\mathcal{B}}$ and $\mathcal{M}_n^{\mathcal{D}_\mathcal{B}}$. We conduct preliminary experiments on ToMe by using CIFAR100 [21] as $\mathcal{D}_\mathcal{A}$, $m = 16$, and $n = 0$. In Table 1, we show the results when using different FGVC datasets [10,20,31,39] as $\mathcal{D}_\mathcal{B}$. We note that by adding $(\mathcal{M}_{16}^{\mathcal{D}_\mathcal{A}} - \mathcal{M}_0^{\mathcal{D}_\mathcal{A}})$, the performance of $\mathcal{M}_0^{\mathcal{D}_\mathcal{B}}$ on $r = 16$ is obviously improved, verifying that *there exist positive transfer of the model gap between two source degrees across different tasks,* which suggests that it is possible to use a universal plugin to model such gap and improve the performance of token compression with unequal source and target degrees on various downstream tasks.

4 Token Compensator

4.1 Arithmetic of Parameter-Efficient Modules

When the compression degrees during training and inference are unequal, the performance degradation of the model occurs due to the differente behaviors

between models with different compression degrees, resulting in distinct local minima in their parameter spaces. To enhance the performance of token compression in such circumstance, especially on the off-the-shelf models, we intend to find a universal plugin to compensate for the gap between models with different compression degrees, assuming that models at two specific compression degrees exhibit similar gaps across different datasets. Supposing we have the plugin $\mathcal{P}_{m\to n}$ which compensates for the gap between model $\mathcal{M}_m$ trained with ToMe $r = m$ and model $\mathcal{M}_n$ trained with ToMe $r = n$, if $\mathcal{M}_m$ is off-the-shelf, we can expect that

$$\mathcal{M}_m \oplus \mathcal{P}_{m\to n} = \mathcal{M}'_n \approx \mathcal{M}_n, \tag{4}$$

in which $\oplus$ denotes architecture-level aggregation, and $\mathcal{M}'_n$ is the synthesized model used for inference with $r = n$.

However, due to the extensive range of the choices in teams of compression degree (*e.g.*, $r \in \{0, 1, ..., 16\}$ in DeiT-B for ToMe), training and storing such plugins for all possible compression degree pairs (*i.e.*, 16×17 choices of (m, n) in Eq. 4) would result in significant training and storage overheads. We address this issue from three perspectives.

First, inspired by parameter-efficient tuning, which uses lightweight modules to describe the gap between pre-trained and fine-tuned models, we also adopt parameter-efficient modules to describe the gap between models with different compression degrees. Concretely, we use LoRA [14] as $\mathcal{P}_{m\to n}$, *i.e.*, for all weight matrix $\mathbf{W} \in \mathbb{R}^{d_1\times d_2}$ in the $\mathcal{M}'_n$ of Eq. 4, we have

$$\begin{cases} \mathbf{W}_{\mathcal{M}'_n} = \mathbf{W}_{\mathcal{M}_m} + s \cdot \mathbf{AB}, & \textit{if } \mathbf{W} \in \{\mathbf{W}_q, \mathbf{W}_v\}, \\ \mathbf{W}_{\mathcal{M}'_n} = \mathbf{W}_{\mathcal{M}_m}, & \textit{otherwise}, \end{cases} \tag{5}$$

in which $\mathbf{W}_q$ and $\mathbf{W}_v$ are weight metrices of query and value transformations, s is a hyper-parameter, and $\mathbf{A} \in \mathbb{R}^{d_1\times h}$ and $\mathbf{B} \in \mathbb{R}^{h\times d_2}$ are LoRA modules. LoRA modules comprise only about 0.1% of the model parameters and do not incur extra computations in inference.

Second, we employ LoRA to estimate the gap between models only at adjacent compression degrees, *i.e.*, $n = m+1$ in Eq. 4. For the cases when $n > m+1$, we simply accumulate all the plugins between m and n, *i.e.*,

$$\mathcal{M}_m \oplus \left(\bigoplus_{i=m}^{n-1} \mathcal{P}_{i\to i+1}\right) = \mathcal{M}'_n \approx \mathcal{M}_n. \tag{6}$$

Third, we assume that the gap between models is invertible, *i.e.*, $\mathcal{P}_{n\to m} = \ominus\mathcal{P}_{m\to n}$. When $n < m$, the plugins are "subtracted", *i.e.*,

$$\mathcal{M}_m \ominus \left(\bigoplus_{i=n}^{m-1} \mathcal{P}_{i\to i+1}\right) = \mathcal{M}'_n \approx \mathcal{M}_n, \tag{7}$$

in which $\ominus$ are the inverse of $\oplus$. To implement $\ominus$, we subtract the LoRA products $\mathbf{AB}$ from the weights instead of adding it.

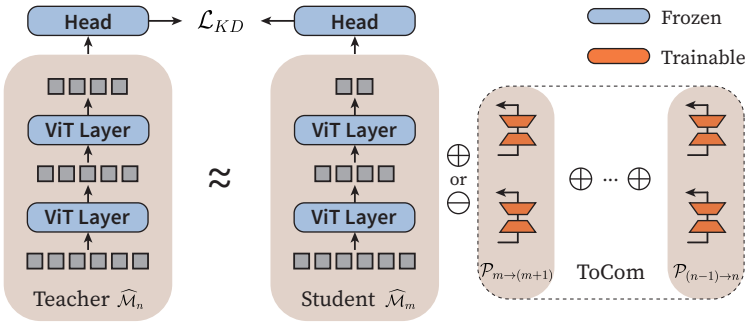

Fig. 4: Illustration of our `ToCom`. `ToCom` is multiple groups of LoRA, which are trained with parameter-efficient self-distillation on pre-training dataset. The teacher model and student model have different token compression degrees which are sampled each step, and `ToCom` is plugged into student model during training.

Now, we only need to train and store the 16 groups of LoRA plugin for ToMe to support all compression degree pairs, whose size is still negligible compared to the ViT backbone. We call the collection of these plugins **To**ken **Com**pensator (`ToCom`). Due to the lightweight nature of `ToCom`, it can be loaded into the RAM with minimal overhead, enabling real-time inference throughput switching.

4.2 Training `ToCom`

As mentioned above, `ToCom` is supposed to be a universal plugin for models tuned on any downstream datasets. To enhance the generalization capability of `ToCom`, we integrate the training of `ToCom` as an extension of the pre-training stage of the ViT backbone. Specifically, we utilize the pre-training data (*e.g.*, ImageNet [6]) to train `ToCom`.

To obtain `ToCom` supporting any compression degree pairs, we propose a self-distillation method for training it. Taking ToMe as an instance, we use $\widehat{\mathcal{M}}_n$ to denote the model obtained by *directly applying ToMe with* $r = n$ *on the off-the-shelf pre-trained model*. The training loss is constructed as follows,

$$\mathcal{L} = \begin{cases} \mathcal{L}_{KD}\left(\widehat{\mathcal{M}}_m \oplus \left(\bigoplus_{i=m}^{n-1} \mathcal{P}_{i\to(i+1)}\right), \widehat{\mathcal{M}}_n\right), & \text{if } n > m \\ \mathcal{L}_{KD}\left(\widehat{\mathcal{M}}_m \ominus \left(\bigoplus_{i=n}^{m-1} \mathcal{P}_{i\to(i+1)}\right), \widehat{\mathcal{M}}_n\right), & \text{if } n < m \end{cases} \tag{8}$$

in which m and n are randomly sampled in each training step satisfying $m \neq n$, and $\mathcal{L}_{KD}$ is the knowledge distillation loss with soft targets, *i.e.*,

$$\mathcal{L}_{KD}(\mathcal{M}_s, \mathcal{M}_t) = \mathrm{KL}(\mathcal{M}_s(\boldsymbol{x}), \mathcal{M}_t(\boldsymbol{x})), \tag{9}$$

where KL denote Kullback–Leibler divergence and $\boldsymbol{x}$ is the inputs.

As illustrated in Fig. 4, during distillation, we freeze all pre-trained parameters including the classification heads, and only update the parameter-efficient

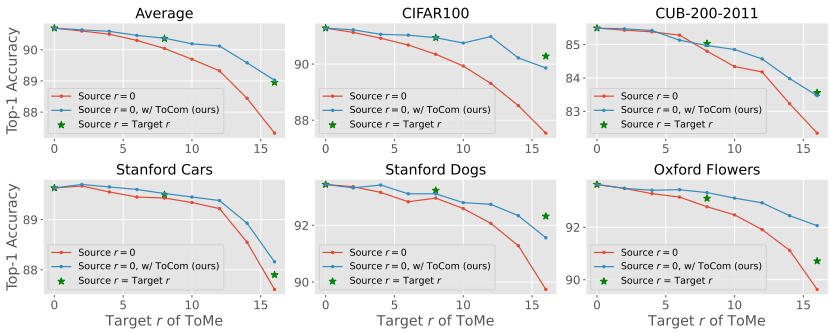

Fig. 5: Results on CIFAR100 and FGVC datasets with source $r = 0$. We also report results when source $r =$ target r as reference.

Table 2: Results on the Natural and Specialized groups of VTAB-1k benchmark.

	Natural									Specialized					
	Cifar100	Caltech101	DTD	Flower102	Pets	SVHN	Sun397	Avg. Acc.	Δ	Camelyon	EuroSAT	Resisc45	Retinopathy	Avg. Acc.	Δ
Target $r = 16$ (1.9× inference speedup)															
Source $r = 0$	56.5	89.6	65.4	85.2	90.2	88.4	40.9	73.7		77.1	90.1	78.6	74.4	80.0	
+ToCom	60.5	90.0	67.2	90.1	91.2	88.5	43.3	75.8	+2.1	83.7	93.6	83.0	74.3	83.6	+3.6
Source $r = 4$	57.5	89.8	64.6	86.6	90.7	88.1	41.1	74.0		78.9	91.6	79.9	73.8	81.0	
+ToCom	60.0	90.2	67.6	89.4	91.2	88.4	42.9	75.7	+1.7	84.5	93.7	83.2	73.7	83.8	+2.8
Source $r = 16$	59.6	89.9	67.5	89.4	90.7	89.4	42.0	75.5		83.2	94.3	83.1	73.8	83.6	
Target $r = 12$ (1.5× inference speedup)															
Source $r = 0$	60.1	90.1	68.0	89.5	91.6	89.2	42.8	75.9		81.2	92.6	82.5	74.8	82.8	
+ToCom	61.4	90.2	68.4	91.3	91.9	89.3	43.8	76.6	+0.7	84.0	94.7	84.7	74.8	84.5	+1.7
Source $r = 4$	60.6	90.3	67.4	90.1	91.8	88.8	42.9	76.0		83.1	93.8	83.1	74.1	83.5	
+ToCom	61.5	90.3	68.0	91.0	92.1	88.9	43.7	76.5	+0.5	85.5	94.3	84.1	74.1	84.5	+1.0
Source $r = 12$	61.0	90.9	67.8	90.8	92.0	90.0	43.1	76.5		84.0	94.6	84.1	73.7	84.1	

ToCom. It is noteworthy that while ToCom requires training on a large-scale dataset, it is lightweight and converges quickly. Thus, its training overhead is negligible compared to the backbone's pre-training.

After training ToCom, we can directly deploy it on off-the-shelf models fine-tuned on any downstream tasks with any source degrees to any target degrees following Eq. 6 and 7. We add or subtract ToCom's LoRA products to the weight of the off-the-shelf models, which are used for inference without further training.

5 Experiments

5.1 Datasets and Setup

We use ImageNet [6] to train ToCom on pre-trained DeiT-B for 10 epochs. In ToCom training, we fix hyper-parameter $s = 0.1$, and choose s from {0.01, 0.02,

Table 3: Results on the Structured group of VTAB-1k benchmark.

	Structured									
	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	Avg. Acc.	Δ
Target $r = 16$ (1.9× inference speedup)										
Source $r = 0$	67.6	56.2	49.4	79.6	78.6	53.3	31.8	36.8	56.7	
+`ToCom`	73.4	57.3	49.6	81.4	78.6	53.5	31.8	37.8	57.9	+1.2
Source $r = 4$	70.5	55.0	49.6	81.3	77.8	53.2	33.2	36.7	57.2	
+`ToCom`	73.6	56.7	50.3	82.0	77.7	53.3	33.3	37.3	58.0	+0.8
Source $r = 16$	77.7	59.6	50.3	80.7	77.8	53.5	33.1	37.5	58.8	
Target $r = 12$ (1.5× inference speedup)										
Source $r = 0$	77.1	57.6	50.2	80.5	79.0	53.7	32.3	38.5	58.6	
+`ToCom`	78.0	58.2	50.5	81.9	79.0	53.8	32.4	38.9	59.1	+0.5
Source $r = 4$	78.2	56.8	50.5	82.0	77.8	53.4	33.9	37.4	58.8	
+`ToCom`	78.6	58.0	50.9	82.8	77.8	53.4	34.0	37.9	59.2	+0.4
Source $r = 12$	79.2	59.3	50.7	82.3	78.0	53.4	33.6	37.9	59.3	

0.05, 0.08, 0.1, 0.12, 0.15} in inference. We use $h = 2$ for each LoRA of `ToCom` such that the total number of parameter of `ToCom` remains merely 1.2M for the DeiT-B with 86M parameters. All other hyper-parameters follow the training recipe of DeiT-B. The training of `ToCom` takes only 1.8 hours on 8×V100 GPUs, which is negligible compared to the 53 hours pre-training of the backbone.

We evaluate `ToCom` on more than 20 downstream datasets, including CIFAR100 [21], 4 fine-grained visual classification (FGVC) datasets, and 19 tasks in VTAB-1k benchmark [47]. The FGVC tasks include CUB-200-2011 [39], Stanford Cars [10], Stanford Dogs [20], and Oxford Flowers [31]. The VTAB-1k benchmark contains tasks in a large range of domains, each of which has 1000 training samples. The 19 tasks of VTAB-1k are further categorized into three groups: Natural, Specialized, and Structured. To obtain the off-the-shelf models on these downstream tasks, we fine-tune CIFAR100, FGVC, and VTAB-1k benchmarks on DeiT-B for 100, 30, and 100 epochs with batch size of 1024, 64, and 64, respectively. Moreover, since recent work has found that PET performs much better than full fine-tuning on VTAB-1k benchmark, we also apply PET when fine-tuning on VTAB-1k, *i.e.*, freezing the pre-trained backbone and updating AdaptFormer [4] with hidden dimension 32. As for CIFAR100 and FGVC datasets, we apply full fine-tuning as default, *i.e.*, updating all backbone parameters. Note that all these datasets are evaluated on the same `ToCom`, and no additional training is required after inserting `ToCom` into the off-the-shelf models, which means we apply pre-trained `ToCom` in a tuning-free manner across all these downstream tasks.

5.2 Inference-Time Acceleration on off-the-Shelf Models

First, we evaluate `ToCom` in the scenario of inference-time acceleration, *i.e.*, applying the models trained with lower source r (or not compression) with higher target r to improve the inference speed.

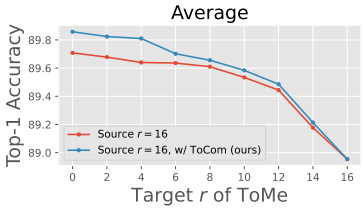

Fig. 6: Average Results on CIFAR100 and FGVC datasets with source $r = 16$.

Table 4: Group-wise average results on VTAB-1k benchmark.

	Natural		Specialized		Structured	
	Avg. Acc.	Δ	Avg. Acc.	Δ	Avg. Acc.	Δ
Source $r = 16$ (1.9× training speedup)						
Target $r = 0$	76.0		84.0		57.8	
+ToCom	76.4	+0.4	84.6	+0.6	59.2	+1.4
Target $r = 4$	76.0		84.2		57.9	
+ToCom	76.3	+0.3	84.6	+0.4	59.2	+1.3
Target $r = 16$	75.5		83.6		58.8	
Source $r = 12$ (1.5× training speedup)						
Target $r = 0$	76.9		84.5		59.1	
+ToCom	77.0	+0.1	84.9	+0.4	59.4	+0.3
Target $r = 4$	76.8		84.6		59.2	
+ToCom	76.9	+0.1	84.9	+0.3	59.5	+0.3
Target $r = 12$	76.5		84.1		59.3	

On CIFAR100 and FGVC datasets, we fix source $r = 0$, *i.e.*, plug ToCom into models trained without ToMe, and evaluate them with target $r \in \{2, 4, 6, 8, 10, 12, 16\}$ with inference speedup from 1.03× to 1.91×. As illustrated in Fig. 5, ToCom improves the average performance of ToMe on all target r. When target $r = 8$ and $r = 16$, ToCom brings 0.5 and 1.7 accuracy gains on average, respectively. Notably, when applying ToCom, the average off-the-shelf performance of the model even matches that of training with source r = target r, demonstrating the significant flexibility brought by ToCom to token compression.

In Table 2 and 3, we list the results on VTAB-1k benchmark when source $r = 0$ or 4 and target $r = 12$ or 16. Under the cases of source $r = 0$ and target $r = 16$, ToCom brings gains of 2.1%, 3.6%, and 1.2% on the average accuracy of the three groups, respectively. We also provide results when source r and target r are equal as a reference. We find that in all four settings of source and target r pairs, ToCom consistently brings significant performance improvements, achieving performance comparable to or better than the results when source r and target r. This once again validates the powerful capability of ToCom to bridge the gap across different r and enhance performance when accelerating inference of off-the-shelf models.

5.3 Training-Time Acceleration

As illustrated in Fig. 3, token compression also accelerates training. We here evaluate ToCom in the scenario when we aim to accelerate model training without prioritizing inference speed. Specifically, we train the model with a higher source r for fast training but employ a lower target r (or not compression) in inference for better performance.

On CIFAR100 and FGVC datasets, we fix source $r = 16$, *i.e.*, 1.9× training speedup, and evaluate them with target $r \in \{0, 2, 4, 6, 8, 10, 12\}$. While on VTAB-1k benchmark, we adopt source $r = 12$ or 16 (1.5× or 1.9× training speedup) and target $r = 0$ or 4. As illustrated in Fig. 6 and Table 4, the

Table 5: Ablation study on the framework of `ToCom`. We report 19-task average results on VTAB-1k benchmark.

	0→16		0→12		16→0	
	Avg. Acc.	Δ	Avg. Acc.	Δ	Avg. Acc.	Δ
Baseline	67.9		70.1		70.0	
Shared LoRA	69.2	+1.3	70.6	+0.5	70.1	+0.1
CLS Loss	**69.9**	+2.0	**70.9**	+0.8	70.6	+0.6
No Inversion	69.8	+1.9	**70.9**	+0.8	70.8	+0.8
`ToCom` (Ours)	**69.9**	+2.0	**70.9**	+0.8	**70.9**	+0.9

Table 6: Ablation study on scaling factor s. We report 19-task average results on VTAB-1k benchmark.

	0→16		0→12		16→0	
	Avg. Acc.	Δ	Avg. Acc.	Δ	Avg. Acc.	Δ
Baseline	67.9		70.1		70.0	
s = 1	**69.9**	+2.0	**70.9**	+0.8	70.8	+0.8
s = 0.1 (Ours)	**69.9**	+2.0	**70.9**	+0.8	**70.9**	+0.9
s = 0.01	**69.9**	+2.0	70.8	+0.7	70.8	+0.8

Table 7: Ablation study on hidden dimension h. We report 19-task average results on VTAB-1k benchmark.

	0→16		0→12		16→0		# Params
	Avg. Acc.	Δ	Avg. Acc.	Δ	Avg. Acc.	Δ	
Baseline	67.9		70.1		70.0		
h = 1	69.7	+1.8	**70.9**	+0.8	70.8	+0.8	0.59M
h = 2 (Ours)	69.9	+2.0	**70.9**	+0.8	**70.9**	+0.9	1.18M
h = 4	69.9	+2.0	**70.9**	+0.8	**70.9**	+0.9	2.36M
h = 8	**70.1**	+2.2	**70.9**	+0.8	**70.9**	+0.9	4.72M

performance of models still benefits from `ToCom`. However, we notice that the improvement from `ToCom` when target $r <$ source r is slighter than that when target $r >$ source r. Moreover, when target $r <$ source r, `ToCom` still exhibits a noticeable gap compared to the models trained with lower source r. This is because models trained with higher source r receive fewer tokens with less diverse information during training which in turn leads to worse performance, despite their faster training speed. However, it is important to note that `ToCom` is a plug-and-play module that does not introduce additional training requirements or inference latency on downstream tasks. Therefore, such performance improvement at no extra cost remains meaningful.

5.4 Ablation Study

To demonstrate the importance of the components of `ToCom`, we conduct ablation experiments. Firstly, to investigate the effects of parameter-efficient module arithmetic and self-distillation loss, we designed the following variants for comparison:

- Shared LoRA: Unlike `ToCom`, which utilizes 16 groups of LoRA with $h = 2$ for arithmetic across different source-target degree pairs, we employ a single group of LoRA with $h = 32$, sharing it among different source-target degree pairs.
- CLS Loss: We use cross-entropy as classification loss instead of self-distillation loss and randomly sample r.
- No Inversion: Unlike `ToCom` which assumes $\mathcal{P}_{n\to m} = \ominus\mathcal{P}_{m\to n}$, we use two groups of LoRA for $\mathcal{P}_{n\to n+1}$ and $\mathcal{P}_{n+1\to n}$, respectively.

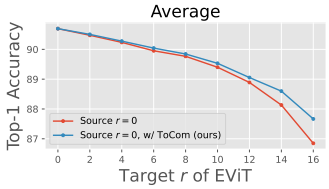

Fig. 7: Performance of EViT on CIFAR100 and FGVC datasets with source $r = 0$.

Table 8: Performance of EViT on VTAB-1k benchmark.

	Natural		Specialized		Structured	
	Avg. Acc.	Δ	Avg. Acc.	Δ	Avg. Acc.	Δ
Target $r = 16$						
Source $r = 0$	73.5		80.4		40.8	
+ToCom	74.6	+1.1	81.7	+1.3	41.8	+1.0
Target $r = 0$						
Source $r = 16$	75.4		84.2		41.6	
+ToCom	75.7	+0.3	84.5	+0.3	43.0	+1.4

We conduct experiments on VTAB-1k benchmark under three setting: (source r, target r) $\in \{(0, 16), (0, 12), (16, 0)\}$. An illustrated in Table 5, we find that: *i*) Shared LoRA performs worse than ToCom across all the three settings, indicating that the LoRA parameters shared among different source-target degree pairs introduce mutual interference. ToCom, on the other hand, avoids this interference by using LoRA to model the gap between adjacent r. *ii*) Classification loss performs comparably to the KD loss when source $r <$ target r. However, it exhibits poorer performance when source $r >$ target r. This is because the classification loss utilizes labels as supervision signals which is equivalent to a strong teacher, and it cannot adapt well to situations where the teacher is weak, such as when source $r >$ target r. *iii*) When we allocate parameters independently for $\mathcal{P}_{n\to n+1}$ and $\mathcal{P}_{n+1\to n}$, the performance slightly decreases despite having twice the number of parameters as ToCom. This indicates that there is positive transfer between $\mathcal{P}_{n\to n+1}$ and $\ominus\mathcal{P}_{n+1\to n}$, suggesting that the assumption of $\mathcal{P}_{n\to m} = \ominus\mathcal{P}_{m\to n}$ is reasonable.

Furthermore, we investigate the impact of the key hyper-parameters of ToCom: LoRA hidden dimension h and scaling factor s, and provide results in Table 6 and 7. In summary, ToCom is not sensitive to s. The performance of ToCom approaches saturation when $h = 2$, so we choose $h = 2$ for better storage-performance trade-off.

5.5 Performance on Other Token Compression Methods

Apart from token merging in ToMe, token pruning is also an important direction for token compression. Although previous research has found that token pruning does not perform as well as token merging on off-the-shelf models [3], we can still apply ToCom to token pruning to improve its performance.

We use EViT [24], a representative token pruning method, as the baseline. EViT uses the attention score of [CLS] tokens to measure the attentiveness of patch tokens and prunes inattentive ones. In the original implementation of EViT, a certain proportion of tokens are pruned from the 4th, 7th, and 11th layers of the ViT. However, we observed that gradually pruning a small number of tokens per layer, similar to ToMe, yields better performance on off-the-shelf models. Therefore, following ToMe, we also prune r tokens per layer for EViT.

Table 9: 19-task average results of DeiT-S on VTAB-1k benchmark.

	Natural		Specialized		Structured	
	Avg. Acc.	Δ	Avg. Acc.	Δ	Avg. Acc.	Δ
Target $r = 16$						
Source $r = 0$	72.8		82.2		55.5	
+ToCom	74.3	+1.5	83.3	+1.1	56.2	+0.7
Target $r = 0$						
Source $r = 16$	74.3		84.1		57.0	
+ToCom	74.5	+0.2	84.4	+0.3	57.1	+0.1

Table 10: 19-task average results of MAE-pretrained ViT-B on VTAB-1k benchmark.

	Natural		Specialized		Structured	
	Avg. Acc.	Δ	Avg. Acc.	Δ	Avg. Acc.	Δ
Target $r = 16$						
Source $r = 0$	60.2		75.3		58.5	
+ToCom	60.7	+0.5	76.6	+1.3	58.9	+0.4
Target $r = 0$						
Source $r = 16$	55.1		79.3		57.8	
+ToCom	55.3	+0.2	79.5	+0.2	58.1	+0.3

As illustrated in Fig. 7 and Table 8, ToCom is also effective for EViT. However, we note that the performance of EViT is much worse than ToMe, especially on Structured group of VTAB-1k. This is because for some downstream tasks, changes in the number of tokens may affect the calculation of the original attention score, but ToMe mitigates this impact through its proportional attention.

5.6 Performance on Other Backbone

We apply ToCom to models with different scales, *i.e.*, DeiT-S, and pre-training object, *i.e.*, masked image modeling. For DeiT-S, we follow the training recipe used for DeiT-B. For ViT-B pre-trained with MAE [13], as it lacks a classification head, we employ L1-norm as the knowledge distillation loss at the feature level of the final layer, *i.e.*, $\mathcal{L}_{KD}(\mathcal{M}_s, \mathcal{M}_t) = \|\mathcal{M}_s(\boldsymbol{x}), \mathcal{M}_t(\boldsymbol{x})\|_1$. As the results shown in Table 9 and 10, ToCom can be generalized to different model scales and pre-training objects. In particular, because the distillation process of ToCom does not require labels, it can be naturally applied to various unsupervised pre-trained backbones.

6 Conclusion

In this paper, we explore the drawbacks of transformer token compression methods applied to downstream tasks. Specifically, we note that the performance of the model is suboptimal when the compression degrees during training and inference are unequal. Based on preliminary experimental results, we speculate that models trained under different compression degrees exhibit a transferable gap between tasks. To address this issue, we propose ToCom, a lightweight pre-trained plugin designed as a plug-and-play compensator to fill this gap and enhance the performance of token compression methods in off-the-shelf models. The effectiveness of ToCom is demonstrated across more than 20 downstream tasks. We believe that the potential of token compression methods has been underestimated in the past, and through ToCom, broader application prospects can be realized.

References

1. Bolya, D., Fu, C., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net (2023), https://openreview.net/pdf?id=JroZRaRw7Eu 1, 2, 4, 5
2. Chavan, A., Shen, Z., Liu, Z., Liu, Z., Cheng, K., Xing, E.P.: Vision transformer slimming: Multi-dimension searching in continuous optimization space. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 4921–4931. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.00488, https://doi.org/10.1109/CVPR52688.2022.00488 1
3. 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. In: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023. pp. 17118–17128. IEEE (2023). https://doi.org/10.1109/ICCV51070.2023.01574, https://doi.org/10.1109/ICCV51070.2023.01574 4, 13
4. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022 (2022), http://papers.nips.cc/paper_files/paper/2022/hash/69e2f49ab0837b71b0e0cb7c555990f8-Abstract-Conference.html 4, 10
5. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net (2023), https://openreview.net/pdf?id=plKu2GByCNW 1
6. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA. pp. 248–255. IEEE Computer Society (2009). https://doi.org/10.1109/CVPR.2009.5206848, https://doi.org/10.1109/CVPR.2009.5206848 8, 9
7. Ding, N., Tang, Y., Han, K., Xu, C., Wang, Y.: Network expansion for practical training acceleration. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023. pp. 20269–20279. IEEE (2023). https://doi.org/10.1109/CVPR52729.2023.01941, https://doi.org/10.1109/CVPR52729.2023.01941 1
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), https://openreview.net/forum?id=YicbFdNTTy 1
9. 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: 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. 396–414. Springer (2022). https://doi.org/10.1007/978-3-031-20083-0_24, https://doi.org/10.1007/978-3-031-20083-0_24 1, 3

10. Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., Fei-Fei, L.: Fine-grained car detection for visual census estimation. In: Singh, S., Markovitch, S. (eds.) Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA. pp. 4502–4508. AAAI Press (2017). https://doi.org/10.1609/AAAI.V31I1.11174, https://doi.org/10.1609/aaai.v31i1.11174 6, 10
11. Han, C., Wang, Q., Cui, Y., Cao, Z., Wang, W., Qi, S., Liu, D.: E^2vpt: An effective and efficient approach for visual prompt tuning. In: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023. pp. 17445–17456. IEEE (2023). https://doi.org/10.1109/ICCV51070.2023.01604, https://doi.org/10.1109/ICCV51070.2023.01604 4
12. Hao, T., Chen, H., Guo, Y., Ding, G.: Consolidator: Mergable adapter with group connections for visual adaptation. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net (2023), https://openreview.net/pdf?id=J_Cja7cpgW 4
13. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.B.: Masked autoencoders are scalable vision learners. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 15979–15988. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.01553, https://doi.org/10.1109/CVPR52688.2022.01553 14
14. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net (2022), https://openreview.net/forum?id=nZeVKeeFYf9 4, 7
15. Ilharco, G., Ribeiro, M.T., Wortsman, M., Schmidt, L., Hajishirzi, H., Farhadi, A.: Editing models with task arithmetic. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net (2023), https://openreview.net/pdf?id=6t0Kwf8-jrj 4
16. Jia, M., Tang, L., Chen, B., Cardie, C., Belongie, S.J., Hariharan, B., Lim, S.: Visual prompt tuning. In: Avidan, S., Brostow, G.J., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIII. Lecture Notes in Computer Science, vol. 13693, pp. 709–727. Springer (2022). https://doi.org/10.1007/978-3-031-19827-4_41, https://doi.org/10.1007/978-3-031-19827-4_41 4
17. Jie, S., Deng, Z.: Fact: Factor-tuning for lightweight adaptation on vision transformer. In: Williams, B., Chen, Y., Neville, J. (eds.) Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023. pp. 1060–1068. AAAI Press (2023). https://doi.org/10.1609/AAAI.V37I1.25187, https://doi.org/10.1609/aaai.v37i1.25187 4
18. Jie, S., Tang, Y., Ding, N., Deng, Z., Han, K., Wang, Y.: Memory-space visual prompting for efficient vision-language fine-tuning. CoRR **abs/2405.05615** (2024). https://doi.org/10.48550/ARXIV.2405.05615, https://doi.org/10.48550/arXiv.2405.05615 4

19. Jie, S., Wang, H., Deng, Z.: Revisiting the parameter efficiency of adapters from the perspective of precision redundancy. In: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023. pp. 17171–17180. IEEE (2023). https://doi.org/10.1109/ICCV51070.2023.01579, https://doi.org/10.1109/ICCV51070.2023.01579 4
20. Khosla, A., Jayadevaprakash, N., Yao, B., Fei-Fei, L.: Novel dataset for fine-grained image categorization: Stanford dogs (2012) 6, 10
21. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009) 6, 10
22. Li, Y., Mao, H., Girshick, R.B., He, K.: Exploring plain vision transformer backbones for object detection. 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 IX. Lecture Notes in Computer Science, vol. 13669, pp. 280–296. Springer (2022). https://doi.org/10.1007/978-3-031-20077-9_17, https://doi.org/10.1007/978-3-031-20077-9_17 1
23. Lian, D., Zhou, D., Feng, J., Wang, X.: Scaling & shifting your features: A new baseline for efficient model tuning. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022 (2022), http://papers.nips.cc/paper_files/paper/2022/hash/00bb4e415ef117f2dee2fc3b778d806d-Abstract-Conference.html 4
24. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Evit: Expediting vision transformers via token reorganizations. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net (2022), https://openreview.net/forum?id=BjyvwnXXVn_ 1, 3, 13
25. Lin, M., Chen, M., Zhang, Y., Shen, C., Ji, R., Cao, L.: Super vision transformer. Int. J. Comput. Vis. **131**(12), 3136–3151 (2023). https://doi.org/10.1007/S11263-023-01861-3, https://doi.org/10.1007/s11263-023-01861-3 3
26. Lin, Y., Zhang, T., Sun, P., Li, Z., Zhou, S.: Fq-vit: Post-training quantization for fully quantized vision transformer. In: Raedt, L.D. (ed.) Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022. pp. 1173–1179. ijcai.org (2022). https://doi.org/10.24963/IJCAI.2022/164, https://doi.org/10.24963/ijcai.2022/164 1
27. Liu, S., Liu, Z., Cheng, K.: Oscillation-free quantization for low-bit vision transformers. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA. Proceedings of Machine Learning Research, vol. 202, pp. 21813–21824. PMLR (2023), https://proceedings.mlr.press/v202/liu23w.html 1
28. Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., Gao, W.: Post-training quantization for vision transformer. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual. pp. 28092–28103 (2021), https://proceedings.neurips.cc/paper/2021/hash/ec8956637a99787bd197eacd77acce5e-Abstract.html 1
29. Luo, S., Tan, Y., Patil, S., Gu, D., von Platen, P., Passos, A., Huang, L., Li, J., Zhao, H.: Lcm-lora: A universal stable-diffusion acceleration module. ArXiv preprint **arxiv:2311.05556** (2023). https://doi.org/10.48550/ARXIV.2311.05556, https://doi.org/10.48550/arXiv.2311.05556 4

30. Marin, D., Chang, J.R., Ranjan, A., Prabhu, A., Rastegari, M., Tuzel, O.: Token pooling in vision transformers for image classification. In: IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023. pp. 12–21. IEEE (2023). https://doi.org/10.1109/WACV56688.2023.00010, https://doi.org/10.1109/WACV56688.2023.00010 1, 4
31. Nilsback, M., Zisserman, A.: Automated flower classification over a large number of classes. In: Sixth Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2008, Bhubaneswar, India, 16-19 December 2008. pp. 722–729. IEEE Computer Society (2008). https://doi.org/10.1109/ICVGIP.2008.47, https://doi.org/10.1109/ICVGIP.2008.47 6, 10
32. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual. pp. 13937–13949 (2021), https://proceedings.neurips.cc/paper/2021/hash/747d3443e319a22747fbb873e8b2f9f2-Abstract.html 3
33. Rebuffi, S., Bilen, H., Vedaldi, A.: Learning multiple visual domains with residual adapters. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 506–516 (2017), https://proceedings.neurips.cc/paper/2017/hash/e7b24b112a44fdd9ee93bdf998c6ca0e-Abstract.html 4
34. Shah, V., Ruiz, N., Cole, F., Lu, E., Lazebnik, S., Li, Y., Jampani, V.: Ziplora: Any subject in any style by effectively merging loras. ArXiv preprint **arxiv:2311.13600** (2023). https://doi.org/10.48550/ARXIV.2311.13600, https://doi.org/10.48550/arXiv.2311.13600 4
35. Shi, D., Tao, C., Rao, A., Yang, Z., Yuan, C., Wang, J.: Crossget: Cross-guided ensemble of tokens for accelerating vision-language transformers. ArXiv preprint **arxiv:2305.17455** (2023). https://doi.org/10.48550/ARXIV.2305.17455, https://doi.org/10.48550/arXiv.2305.17455 4
36. Strudel, R., Pinel, R.G., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. pp. 7242–7252. IEEE (2021). https://doi.org/10.1109/ICCV48922.2021.00717, https://doi.org/10.1109/ICCV48922.2021.00717 1
37. Tang, Y., Han, K., Wang, Y., Xu, C., Guo, J., Xu, C., Tao, D.: Patch slimming for efficient vision transformers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 12155–12164. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.01185, https://doi.org/10.1109/CVPR52688.2022.01185 3
38. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 10347–10357. PMLR (2021), http://proceedings.mlr.press/v139/touvron21a.html 1, 6
39. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011) 6, 10

40. Wei, S., Ye, T., Zhang, S., Tang, Y., Liang, J.: Joint token pruning and squeezing towards more aggressive compression of vision transformers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023. pp. 2092–2101. IEEE (2023). https://doi.org/10.1109/CVPR52729.2023.00208, https://doi.org/10.1109/CVPR52729.2023.00208 1, 4
41. Wu, X., Zeng, F., Wang, X., Wang, Y., Chen, X.: PPT: token pruning and pooling for efficient vision transformers. ArXiv preprint **arxiv:2310.01812** (2023). https://doi.org/10.48550/ARXIV.2310.01812, https://doi.org/10.48550/arXiv.2310.01812 4
42. Xu, Y., Zhang, Z., Zhang, M., Sheng, K., Li, K., Dong, W., Zhang, L., Xu, C., Sun, X.: Evo-vit: Slow-fast token evolution for dynamic vision transformer. In: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022. pp. 2964–2972. AAAI Press (2022). https://doi.org/10.1609/AAAI.V36I3.20202, https://doi.org/10.1609/aaai.v36i3.20202 1, 3
43. Yin, H., Vahdat, A., Álvarez, J.M., Mallya, A., Kautz, J., Molchanov, P.: A-vit: Adaptive tokens for efficient vision transformer. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 10799–10808. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.01054, https://doi.org/10.1109/CVPR52688.2022.01054 1, 3
44. Yoo, S., Kim, E., Jung, D., Lee, J., Yoon, S.: Improving visual prompt tuning for self-supervised vision transformers. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA. Proceedings of Machine Learning Research, vol. 202, pp. 40075–40092. PMLR (2023), https://proceedings.mlr.press/v202/yoo23a.html 4
45. Yu, L., Xiang, W.: X-pruner: explainable pruning for vision transformers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023. pp. 24355–24363. IEEE (2023). https://doi.org/10.1109/CVPR52729.2023.02333, https://doi.org/10.1109/CVPR52729.2023.02333 1
46. Zaken, E.B., Goldberg, Y., Ravfogel, S.: Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022. pp. 1–9. Association for Computational Linguistics (2022). https://doi.org/10.18653/V1/2022.ACL-SHORT.1, https://doi.org/10.18653/v1/2022.acl-short.1 4
47. Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A.S., Neumann, M., Dosovitskiy, A., Beyer, L., Bachem, O., Tschannen, M., Michalski, M., Bousquet, O., Gelly, S., Houlsby, N.: The visual task adaptation benchmark. ArXiv preprint **arxiv:1910.04867** (2019), http://arxiv.org/arxiv:1910.04867 10
48. Zhang, J.O., Sax, A., Zamir, A., Guibas, L.J., Malik, J.: Side-tuning: A baseline for network adaptation via additive side networks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part III. Lecture

Notes in Computer Science, vol. 12348, pp. 698–714. Springer (2020). https://doi.org/10.1007/978-3-030-58580-8_41, https://doi.org/10.1007/978-3-030-58580-8_41 4

49. Zhang, J., Chen, S., Liu, J., He, J.: Composing parameter-efficient modules with arithmetic operation. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023), http://papers.nips.cc/paper_files/paper/2023/hash/299a08ee712d4752c890938da99a77c6-Abstract-Conference.html 4