CAIT: Triple-Win Compression Toward High Accuracy, Fast Inference, and Favorable Transferability for ViTs

Ao Wang, *Graduate Student Member, IEEE*, Hui Chen, Zijia Lin, Sicheng Zhao, *Senior Member, IEEE*, Jungong Han, *Senior Member, IEEE*, and Guiguang Ding, *Senior Member, IEEE*

***Abstract*—Vision Transformers (ViTs) have emerged as state-of-the-art models for various vision tasks recently. However, their heavy computation costs remain daunting for resource-limited devices. To address this, researchers have dedicated themselves to compressing redundant information in ViTs for acceleration. However, existing approaches generally sparsely drop redundant image tokens by token pruning or brutally remove channels by channel pruning, leading to a sub-optimal balance between model performance and inference speed. Moreover, they struggle when transferring compressed models to downstream vision tasks that require the spatial structure of images, such as semantic segmentation. To tackle these issues, we propose CAIT, a joint compression method for ViTs that achieves a harmonious blend of high accuracy, fast inference speed, and favorable transferability to downstream tasks. Specifically, we introduce an asymmetric token merging (ATME) strategy to effectively integrate neighboring tokens. It can successfully compress redundant token information while preserving the spatial structure of images. On top of it, we further design a consistent dynamic channel pruning (CDCP) strategy to dynamically prune unimportant channels in ViTs. Thanks to CDCP, insignificant channels in multi-head self-attention modules of ViTs can be pruned uniformly, significantly enhancing the model compression. Extensive experiments on multiple benchmark datasets show that our proposed method can achieve state-of-the-art performance across various ViTs.**

***Index Terms*—Model compression, vision transformer, channel pruning, token pruning.**

I. Introduction

RECENTLY, the field of computer vision has witnessed significant progress with the emergence of Vision Transformer (ViT) [1] and its variants [2], [3], [4], [5], [6]. These models have demonstrated exceptional performance on various vision tasks [7], [8], [9], [10], [11], [12], surpassing the state-of-the-art convolutional neural networks (CNNs). Building upon the success of transformers [13], [14], [15] in natural language processing (NLP), scaling ViTs has become a key priority [16], [17], [18], [19]. This has led to the development of various vision foundation models, such as ViT-22B [19] and SAM [20]. However, the high computation and memory costs of these models have posed significant challenges [21], [22], limiting their practical applications, especially on resource-limited devices. Therefore, compressing and accelerating ViTs are critical for making them viable for real-world applications [23], [24], [25].

Early attempts follow previous experiences in compressing CNN models, which aim to reduce redundant channels in a structured manner. They usually adopt a pruning-then-finetuning scheme via sparse learning [26], taylor expansion [27], [28], or collaborative optimization [24]. Dynamic channel pruning [21], [23] is also applied for ViTs to identify unimportant channels during fine-tuning, achieving remarkable performance. Based on the intuition that many tokens encode less important or similar information, such as background details [22], [25], [29], [30], recent works investigate to prune redundant tokens to accelerate the transformer computation. For example, DynamicViT [25] eliminates tokens based on their predicted importance scores. Considering that channel pruning and token pruning compress redundant information from model level (*i.e.*, parameters) and data level (*i.e.*, tokens), respectively, conducting them separately may lead to an excessive reduction on one level while neglecting the redundancy on the other level, which compromises overall model quality [21], [28]. Thus, recently, there have been works utilizing token pruning and channel pruning for collaborative compression of ViTs, achieving state-of-the-art performance [28], [31].

It is crucial to ensure that a compressed model can perform effectively and efficiently in practical applications. Therefore, existing works often prioritize compressing a pretrained model based on the principles of high accuracy and fast inference. High accuracy ensures that the capacity of the compressed model remains comparable to the original model. Fast inference, on the other hand, guarantees that the compressed model can make predictions quickly, which is particularly important in resource-constrained environments where low latency is essential. However, we argue that relying solely on accuracy and

Received 6 April 2025; accepted 19 September 2025. Date of publication 1 October 2025; date of current version 9 January 2026. This work was supported by the National Natural Science Foundation of China under Grant 62525103, Grant 624B2082, Grant 62271281, Grant 62441235, and Grant 62571294. Recommended for acceptance by V. Morariu. *(Corresponding authors: Hui Chen; Guiguang Ding.)*

Ao Wang and Guiguang Ding are with BNRist, Tsinghua University, Beijing 100190, China, and also with the School of Software, Tsinghua University, Beijing 100190, China (e-mail: wanga24@mails.tsinghua.edu.cn; dinggg@tsinghua.edu.cn).

Hui Chen and Sicheng Zhao are with BNRist, Tsinghua University, Beijing 100190, China (e-mail: jichenhui2012@gmail.com; schzhao@tsinghua.edu.cn).

Zijia Lin is with the School of Software, Tsinghua University, Beijing 100190, China (e-mail: linzijia07@tsinghua.org.cn).

Jungong Han is with the Department of Automation, Tsinghua University, Beijing 100190, China (e-mail: jungonghan77@gmail.com).

Digital Object Identifier 10.1109/TPAMI.2025.3616854

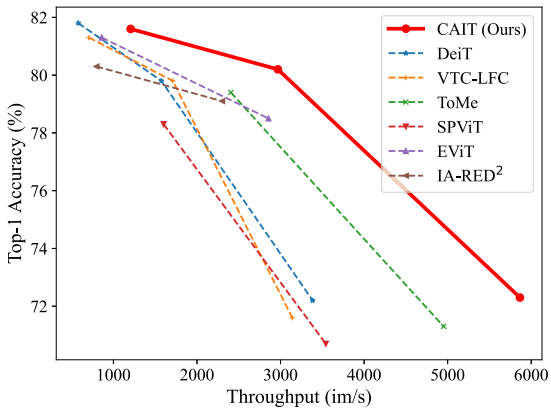

Fig. 1. Comparison of throughput and accuracy between CAIT (Ours) and other models. The top-1 accuracy is tested on ImageNet-1 K and the throughput is evaluated on a NVIDIA RTX-3090 GPU with a batch size of 256.

TABLE I
COMPARISON OF THE PROPOSED CAIT WITH SEVERAL STATE-OF-THE-ART METHODS

Method	High accuracy	Fast inference	Favorable transferability
EViT [30]	✓	✓	✗
ToMe [34]	✓	✓	✗
Evo-ViT [41]	✗	✓	✓
IA-RED^2 [33]	✓	✓	✗
dTPS [35]	✓	✓	✗
SPViT [43]	✓	✗	✓
VTC-LFC [28]	✓	✗	✗
CAIT (ours)	✓	✓	✓

inference efficiency does not guarantee favorable transferability, *i.e.*, enjoying the same remarkable performance in various downstream tasks. Existing works predominantly focus on achieving high accuracy in image classification on the ImageNet [32] dataset without sufficient consideration for performance after transferring to downstream tasks. Therefore, they generally struggle to achieve triple-wins among the accuracy, the inference efficiency, and the transferability. For example, advanced channel pruning methods [21], [23] directly remove attention heads without deeper exploration of sparsity in the multi-head self-attention (MHSA) module, easily causing over-pruning of parameters in MHSA. Unstructured token pruning methods [22], [25], [28], [30], [33], [34], [35] usually drop redundant tokens sparsely, resulting in the disruption of the spatial structure of images, and thus leading to harmful impacts in downstream tasks heavily reliant on visual structure, such as semantic segmentation [36], [37], instance segmentation [38], [39], visual enhancement [40], and so on. Structured token pruning [41], [42] methods can maintain the spatial structure of images. However, they obtain inferior performance to unstructured ones [28], [35]. State-of-the-art methods [28], [31], which combine token pruning and channel pruning, simply adopt principles of unstructured token pruning and pruning-then-finetuning channel pruning. They still fail to achieve a good balance among the performance, inference speed, and the transferability. For example, VTC-LFC [28] enjoys state-of-the-art performance but with slow inference speed and limited transferability.

In this work, we aim to deliver a triple-win **C**ompression method, dubbed **CAIT**, which achieves **high A**ccuracy, **fast I**nference speed, and **favorable T**ransferability all at once for pretrained ViTs. CAIT comprises two key strategies: the asymmetric token merging (ATME) strategy and the consistent dynamic channel pruning (CDCP) strategy. As shown in Fig. 2(a) and Fig. 2(b), rather than sparsely leaving out redundant tokens, ATME utilizes horizontal token merging and vertical token merging to integrate neighboring tokens. When adapting the model for downstream vision tasks after token pruning, the feature map of patches can be simply upsampled to restore the original spatial integrity. Therefore, ATME can effectively reduce the number of tokens while maintaining a complete spatial structure. Meanwhile, as shown in Fig. 2(c) and Fig. 2(d), CDCP adopts the dynamic channel pruning approach and encourages the structured sparsity model gradually by selecting unimportant channels periodically, avoiding the irreversible removal of important channels in previous works. It also employs head-level consistency and attention-level consistency to perform fine-grained compression for all modules. As a result, unimportant channels in ViTs, including the MHSA modules, can be uniformly removed, enabling fast parallel computing and thus enhancing the model compression. Besides, CDCP has no impact on the spatial structure of the image patches and preserves their spatial integrity. Combined with its high performance after pruning, it also facilitates the transferability of models. Table I presents the comparison of our proposed CAIT with other methods.

The proposed joint compression method can be seamlessly applied to prune well pretrained ViTs through a single fine-tuning process. Thanks to ATME and CDCP, redundant tokens and channels in pretrained ViTs can be simultaneously compressed, resulting in a considerable boost of computation efficiency without performance degradation. More importantly, the spatial structure of images are largely preserved during pruning, offering significant benefits for transferring to downstream tasks. Experiments on ImageNet show that our method can significantly outperform the state-of-the-art methods in terms of both the performance and the inference speed, as shown in Fig. 1. Notably, our pruned DeiT-Tiny and DeiT-Small can achieve speedups of 1.7× and 1.9×, respectively, without any compromise in performance. Our compressed DeiT-Base model achieves an impressive speedup of 2.1× with a negligible 0.2% accuracy decline. In addition, when adapting our accelerated backbones to the downstream vision task of semantic segmentation, our method can provide up to 1.31× faster overall throughput without sacrificing performance, demonstrating its strong transferability.

In summary, our contributions are four-fold:

- Beyond high accuracy and fast inference speed, we propose the incorporation of a novel principle: favorable transferability when designing compression algorithms. Thus, we present a joint compression method, dubbed CAIT,

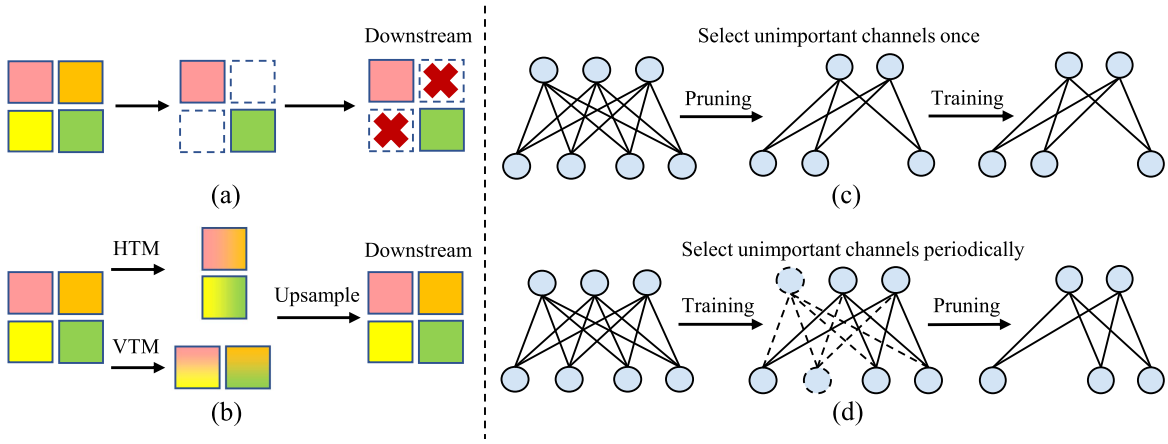

Fig. 2. The comparisons between CAIT and others. (a) indicates that previous token pruning methods usually sparsely drop the tokens and disrupt the complete spatial structure when transferring to downstream tasks. (b) shows that ATME leverages horizontal token merging (HTM) and vertical token merging (VTM) to prune tokens while maintaining the spatial integrity, where image feature maps can be easily upsampled for downstream tasks. (c) denotes that previous channel pruning methods usually suffer from irreversible removal of important channels due to the direct pruning. (d) presents that instead of directly modifying the model structure like previous works, CDCP dynamically determines the importance of channels periodically and encourages structured sparsity network gradually during training, which can thus recover the important channels.

towards high accuracy, fast inference speed, and favorable transferability all at once for ViTs.

- We propose an asymmetric token merging strategy that effectively reduces the number of tokens while preserving complete spatial structure of images. It results in efficient models that are highly suitable for downstream tasks.
- We introduce consistent dynamic channel pruning strategy that achieves dynamic fine-grained compression optimization for all modules, further enhancing compression.
- Extensive experiments on various ViTs show that our method consistently achieves state-of-the-art results in terms of accuracy and inference speed, demonstrating its effectiveness. Experiments on transferring pruned ViTs to various downstream tasks verify the excellent transferability of our proposed method.

The subsequent sections of the paper are structured as follows. In Section II, we conduct a thorough examination of related literature. Subsequently, in Section III, we present our joint compression method CAIT, including the details of ATME and CDCP. In Section IV, we evaluate the performance of CAIT and conduct comprehensive analyses on various benchmark datasets. Finally, we conclude in Section V.

II. Related Work

Vison Transformer. Inspired by remarkable achievements of transformer models [44] in natural language processing, Vision Transformer (ViT) [1] was introduced to leverage the pure transformer architecture for vision tasks. With large-scale training data, ViT has shown outstanding performance on various image classification benchmarks, surpassing state-of-the-art convolutional neural networks (CNNs) [1], [6], [45]. Since then, many follow-up variants of ViT have been proposed [46], [47], [48], [49], [50], [51], [52]. For example, DeiT [6] presents a data efficiency training strategy for ViT by leveraging the teacher-student architecture. In addition to image classification, many novel ViTs have also achieved remarkable performance in various other vision tasks, such as object detection [53], [54], [55], [56], image retrieval [57], [58], semantic segmentation [59], [60], [61], [62], image reconstruction [12], [63], and 3D point cloud processing [64]. However, despite impressive performance, the intensive computation costs and memory footprint greatly hinder the efficient deployment of ViTs for practical applications [21]. This naturally calls for the study of efficient ViTs, including token pruning [25], [65], channel pruning [26], [28], and weights sharing [66], *etc*..

Token Pruning for ViTs. Token pruning for ViTs aims to reduce the number of processed tokens to accelerate the inference speed [25], [30]. For example, DynamicViT [25] removes less important tokens by evaluating their significance via a MLP based prediction module. Additionally, SiT [67] proposes a token slimming module by dynamic token aggregation, meanwhile leveraging a feature distillation framework to recalibrate the unstructured tokens. Although achieving promising performance, most existing token pruning methods select tokens in an unstructured manner [22], [25], [28], [30], [33], [34], [35], *i.e.*, discarding redundant tokens sparsely, which inevitably damages the integrity of spatial structure. This greatly hinders the accelerated model transferred to downstream vision tasks depending on a complete spatial structure, such as semantic segmentation. Besides, existing structured token pruning methods [41], [42] fail to maintain dense pixel information or discriminative token features, still leading to the limited transferability.

Channel Pruning for ViTs. Channel pruning for ViTs involves removing redundant parameters to obtain a more lightweight model [21]. For example, NViT [27] proposes to greedily remove redundant channels by estimating their importance scores with the Taylor-based scheme. Additionally, SAViT [24] explores collaborative pruning by integrating essential structural-aware interactions between different components in ViTs. However, most channel pruning methods typically follow a two-stage approach and thus suffer from limitations stemming from the

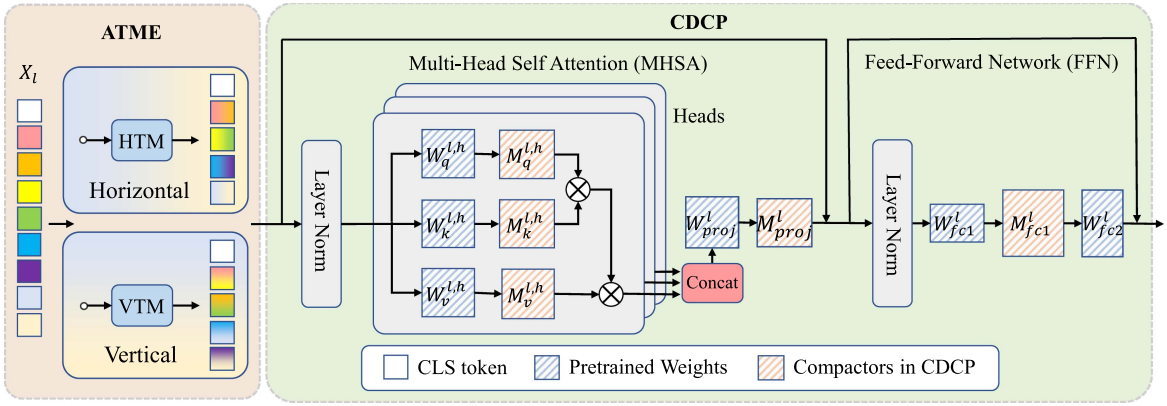

Fig. 3. The overview of our proposed joint compression method for ViTs. We design an asymmetric token merging (ATME) strategy with horizontal token merging (HTM) and vertical token merging (VTM) to prune tokens while preserving their spatial integrity. Consistent dynamic channel pruning (CDCP) is further introduced to enable dynamic fine-grained compression optimization for all learnable weights with minimal performance degradation.

pruning process, in which the irreversible pruning may result in irreparable loss of important channels being erroneously pruned [24], [27], [28]. Besides, existing dynamic channel pruning methods for ViTs [21], [23] have limitations when it comes to the fine-grained pruning of MHSA modules due to the self-attention dimension constraints, thus leading to sub-optimal model quality and performance after compression.

Joint Compression for ViTs. Joint compression for ViTs aims to utilize token pruning and channel pruning for collaborative compression. It reduces both redundant data-level (*i.e.*, tokens) and model-level (*i.e.*, parameters) information in ViTs, achieving state-of-the-art performance [28], [31]. For example, VTC-LFC [28] presents bottom-up cascade pruning framework to jointly compress channels and tokens that are less effective to encode low-frequency information. [31] proposes a statistical dependence based pruning criterion to identify deleterious tokens and channels jointly. However, existing joint compression methods simply adopt the unstructured token pruning and pruning-then-finetuning channel pruning principles, failing to achieve superiority over model performance, the inference speed and transferability at the same time.

III. Methodology

A. Preliminary

We first introduce the necessary notations. The ViT model is composed of L stacked transformer blocks. As shown in Fig. 3, each transformer block comprises a multi-head self-attention (MHSA) module and a feed-forward network (FFN) module. For ease of explanation, we omit the CLS token for all input notations, because the CLS token is not involved in the token pruning. Given an input image, it is split into a sequence of tokens by patchify operation, which is then fed into transformer blocks to extract visual features. We use $X_l \in R^{N_l \times C}$ to denote the tokens in the l-th block, where N_l is the number of tokens and C is the dimension of token's feature. In the l-th transformer block, MHSA is parameterized by $W_q^{l,h}, W_k^{l,h}, W_v^{l,h} \in R^{C \times D}$ and $W_{proj}^l \in R^{C \times C}$, where h denotes the index of head and D is the head dimension. It can be formulated by:

$$\begin{aligned} \text{MHSA}(X_l) &= \text{CONCAT}(\text{head}_0, \ldots, \text{head}_h, \ldots) W_{proj}^l, \\ \text{head}_h &= \text{softmax}\left(\frac{(X_l W_q^{l,h})(X_l W_k^{l,h})^T}{\sqrt{D}}\right)(X_l W_v^{l,h}). \end{aligned} \tag{1}$$

Similarly, FFN is parameterized by $W_{fc1}^l \in R^{C \times 4C}$ and $W_{fc2}^l \in R^{4C \times C}$. In this work, we aim to simultaneously reduce the token number N_l and prune redundant channels in all parameters through the proposed joint compression method, as illustrated by Fig. 3.

B. Asymmetric Token Merging

Most existing token pruning methods focus solely on image classification, and generally reduce the number of tokens in an unstructured manner [22], [25], [28], [30], [33], [34], [35], *i.e.*, by discarding tokens sparsely. However, although remarkable success has been achieved, sparsely wiping out redundant tokens inevitably disrupts the spatial integrity of images. Thus, the compressed ViT models are not suitable for downstream vision tasks that depend on a complete spatial structure, such as semantic segmentation, which significantly restricts their transferability. Besides, existing structured token pruning methods [41], [42] suffer from significant loss of dense information or token features, resulting in inferior performance during transferring. Here, we present an asymmetric token merging strategy to effectively accelerate ViTs, meanwhile maintaining the strong transferability of ViTs. Specifically, we introduce two basic token merging operations to integrate token features while preserving spatial integrity.

Horizontal Token Merging (HTM): As shown in Fig. 4.(a), for a sequence of tokens $X_l \in R^{N_l \times C}$ to be processed, we first reshape it to the shape of feature maps, *i.e.*, $\overline{X}_l \in R^{H \times W \times C}$, where H and W are the height and width of features maps. Then, we group and concatenate two adjacent tokens horizontally, by which we can obtain $\widehat{X}_l \in R^{H \times \frac{W}{2} \times 2C}$, where $2C$ is the feature dimension after concatenation. We leverage a lightweight linear

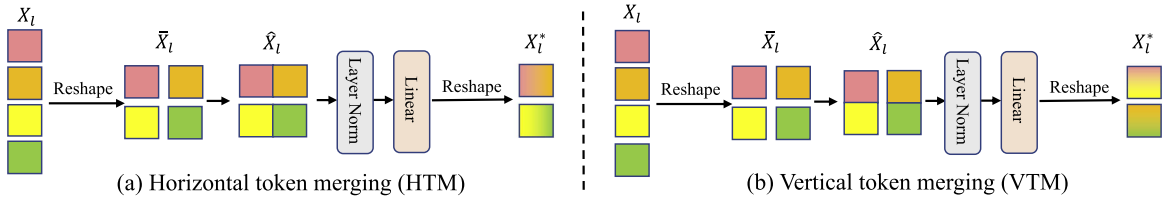

Fig. 4. The proposed asymmetric token merging strategy prunes tokens horizontally or vertically.

layer to effectively fuse the features of grouped tokens by $\widetilde{X}_l = \mathsf{Linear}(\mathsf{LayerNorm}(\widehat{X}_l)) \in R^{H\times\frac{W}{2}\times C}$. We then reshape it back to obtain a sequence of tokens, *i.e.*, $X_l^* \in R^{\frac{N_l}{2}\times C}$.

Vertical Token Merging (VTM): As shown in Fig. 4.(b), similar to horizontal token merging, after obtaining $\overline{X}_l \in R^{H\times W\times C}$, we group and concatenate two adjacent tokens along the vertical direction, ending up with $\widehat{X}_l \in R^{\frac{H}{2}\times W\times 2C}$. Similarly, we can obtain the fused token features by $\widetilde{X}_l = \mathsf{Linear}(\mathsf{LayerNorm}(\widehat{X}_l)) \in R^{\frac{H}{2}\times W\times C}$. Then, we can derive the final token features $X_l^* \in R^{\frac{N_l}{2}\times C}$ after decreasing the number of tokens by reshaping.

By leveraging these two basic operations, we can obtain asymmetric feature maps through asymmetric merging in ViTs. Besides, both operations are generic and plug-and-play. We can seamlessly integrate them into ViTs without complicated hyper-parameter tuning. Following [22], [25], [30], [35], we hierarchically alternatively utilize horizontal and vertical token merging before MHSA through the whole network for the token pruning. Specifically, we first prioritize the strategy of uniformly dividing layers for pruning based on the expected FLOPs reduction. For example, if the target FLOPs reduction ratio for token pruning for a 12-layer DeiT-Small is 43.3%, we initially select the 5-th layer and 9-th layer which are evenly sampled to perform HTM and VTM, respectively, resulting in a FLOPs reduction of 41.1%. Either HTM first or VTM first makes a negligible difference according to our results. Subsequently, minor adjustments are made to the positions of the pruning layers, to align better with the desired ratio of FLOPs reduction. For example, we then adjust the pruning layer from the 9-th to the 8-th layer, resulting in an exact FLOPs reduction of 43.3%. In this way, we can progressively reduce the number of tokens in ViTs while still maintaining the integrity of spatial structure for image features.

C. Consistent Dynamic Channel Pruning

Dynamic Channel Pruning. Two-stage channel pruning methods involve pruning a pretrained model and subsequently fine-tuning the pruned model [68], [69]. However, these methods have limitations due to the pruning process, in which irreversible pruning can lead to the unintended removal of crucial channels and cause irreparable loss. In contrast, dynamic channel pruning dynamically determines the importance of channels during fine-tuning and encourages unimportant channels to gradually approach zero importance [70], [71], [72]. After fine-tuning, channels converging to zero importance are eliminated, resulting in the compressed model. In such a way, important channels can

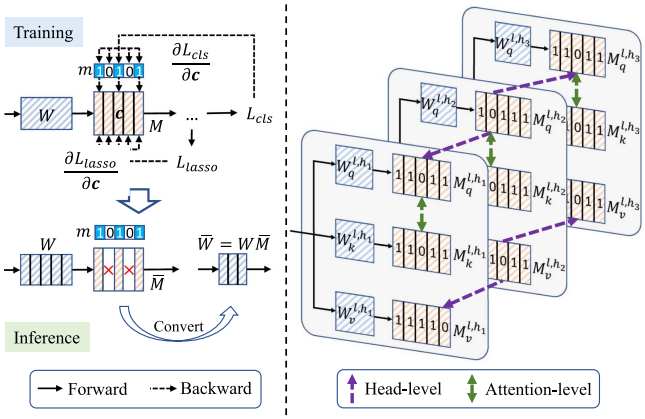

Fig. 5. Left: Framework of the compactor. Right: Consistencies for pruning MHSA.

be recovered during training, thus leading to improved overall performance. Previous works for dynamic channel pruning focus on the design of metrics for deciding the importance of channels. Among them, compactor-based method [70] achieves the state-of-the-art performance for CNN pruning. Here, we propose a consistent dynamic channel pruning strategy based on [70] to perform fine-grained compression optimization for all modules in ViTs.

As shown in Fig. 5, following [70], we insert a compactor, which is a learnable transformation matrix, for each parameter in ViT. For the generality of description, we denote a compactor and its preceding weight as M and W, respectively, if not specified. Otherwise, we add super/sub-scripts to them to indicate their positions. For example, $M_q^{l,h} \in R^{D\times D}$ denotes the compactor corresponding to $W_q^{l,h}$ for the h-th head in the l-th block. Intuitively, in the compactor M, each column $\boldsymbol{c} \in M$ corresponds to one output channel of W. The norm of $\boldsymbol{c}$ can revel the importance of channels for W. Therefore, during training, we adopt the group lasso regularizer [73], [74] to dynamically push channels of M to be sparse, *i.e.*, $L_{lasso} = ||\boldsymbol{c}||_2$. As [70], we introduce a mask variable $m \in \{0, 1\}$ for each $\boldsymbol{c}$ to indicate the corresponding channel is pruned, *i.e.*, $m = 0$, or not, *i.e.*, $m = 1$. We update the gradient of $\boldsymbol{c}$ manually by

$$\nabla c = m\frac{\partial L_{cls}}{\partial \boldsymbol{c}} + \lambda\frac{\partial L_{lasso}}{\partial \boldsymbol{c}}, \tag{2}$$

where L_{cls} is the classification objective and λ is a hyper-parameter. For every several iterations, we set the masks of $\boldsymbol{c}$ with the lowest norm values to 0 for encouraging the unimportant channels to approach zero importance. After training, we can

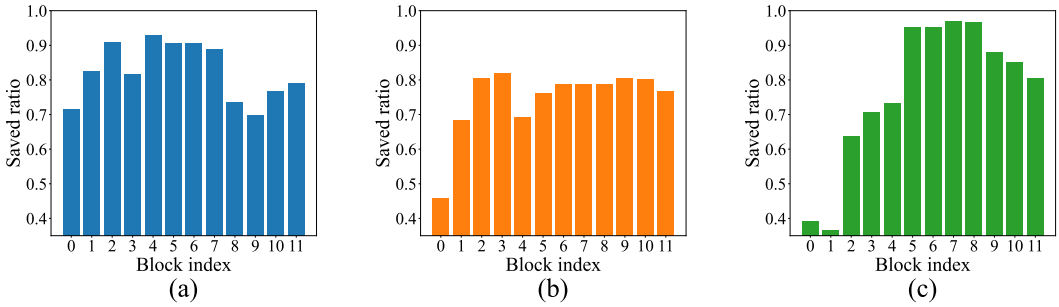

Fig. 6. Architecture of pruned (a) DeiT-Tiny, (b) DeiT-Small, and (c) DeiT-Base models.

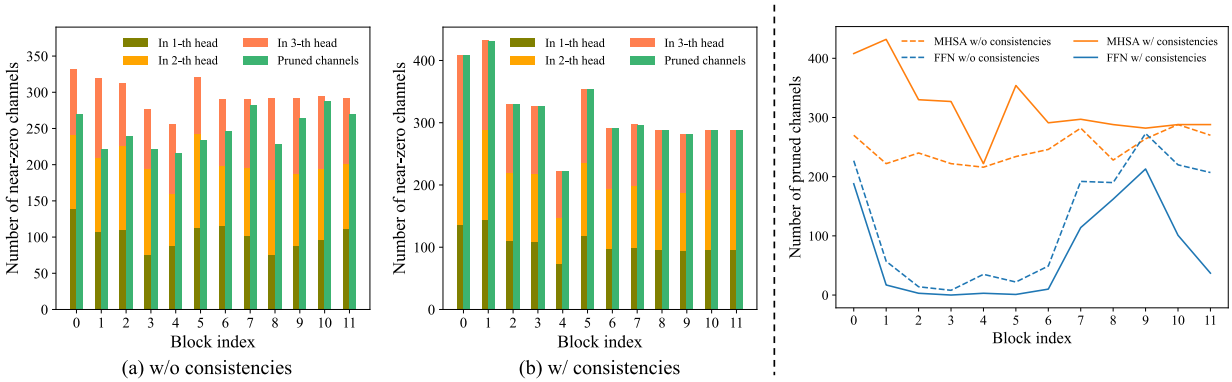

Fig. 7. Near-zero importance channels VS. Pruned channels in MHSA (left). Pruned channels in MHSA and FFN (right).

eliminate redundant (converging to zero importance) channels in M, ending up with a pruned compactor $\overline{M}$. Then, a pruned weight, denoted as $\overline{W}$, can be derived by $\overline{W} = W\overline{M}$.

However, the vanilla compactor pruning applies a global selection criteria to identify unimportant channels and encourages them to approach zero importance. It may result in

1) imbalanced sparse ratios of channels among heads after fine-tuning. Therefore, only the minimum ratio of near-zero importance channels across heads can be removed for efficient parallel self-attention computation; and
2) inconsistent channel importance between $W_q^{l,h}$ and $W_k^{l,h}$ which leads to different sparse outcomes. For example, one channel's importance may be close to zero importance in the query while its corresponding one in the key is not. Therefore, only the channels with zero importance in both the query and key can be pruned for error-free self-attention computation.

However, in such a way, a substantial number of near-zero importance channels are retained in the compressed model (Fig. 7), degrading the performance (Table XI).

Here, to address these issues, we introduce the head-level consistency and attention-level consistency for pruning ViTs, as shown in Fig. 5. Specifically, we first formulate the importance score of channel $\boldsymbol{c}$ as $s = ||\boldsymbol{c}||_2$. For channels with the same position in $M_q^{l,h}$ and $M_k^{l,h}$, their scores will be normalized as the mean of the corresponding original scores. Then, we can obtain a global set S, which contains scores of all channels in compactors. Meanwhile, we can derive local score sets, *i.e.*, $S_q^{l,h}$, $S_k^{l,h}$ and $S_v^{l,h}$, for compactors in MHSA, *i.e.*, $M_q^{l,h}$, $M_k^{l,h}$ and $M_v^{l,h}$, respectively. We initialize an empty set P to record unimportant channels for sparsity via (2). Then, we iteratively find unimportant channels and add them to P until a pre-defined FLOPs reduction ratio r_{target} is achieved.

Head-level Consistency: We use this strategy to ensure that the ratios of unimportant channels across heads remain the same. It guarantees that after pruning, the same number of channels remain per head, allowing the MHSA module to be computed in parallel for fast inference. We take $M_q^{l,h}$ as an example. As shown in Algorithm 1, suppose we have selected a channel $\boldsymbol{c} \in M_q^{l,h}$. For other head $h' \neq h$, we remove the smallest score from the local score set $S_q^{l,h'}$ (Algorithm 1) and add its corresponding channel $\boldsymbol{c}' \in M_q^{l,h'}$ to P (Algorithm 1). At each iteration of channel selection of $\boldsymbol{c}$ in head h, we can make sure that the same channels can be added into P for all heads, which will gradually become sparse during fine-tuning. After pruning, the same number of channels in different heads will thus be maintained, resulting in the same shape for each head.

Attention-level Consistency: It is designed to encourage consistent behavior for channels in the same position between $M_q^{l,h}$ and $M_k^{l,h}$. As shown in Algorithm 2, if the i-th channel in $M_q^{l,h}$, *i.e.*, $\boldsymbol{c}$, is added into P, the i-th channel in $M_k^{l,h}$, *i.e.*, $\boldsymbol{c}'$, is added into P as well, no matter how large its importance score is. Channels in $M_k^{l,h}$ will be managed in the same way. As a result, channels in query and key will be encouraged sparsity simultaneously. Therefore, after finetuning, channels with the same position in query and key will be of similar

Algorithm 1: Head-Level Consistency.

Input: Selected channel $\boldsymbol{c} \in M_q^{l,h}$, score sets for each head $\{S_q^{l,1}, ..., S_q^{l,h}, ...\}$
Output: Expanded channels P for sparsity

$S_q^{l,h} \leftarrow S_q^{l,h} \setminus \{s_{\boldsymbol{c}}\}$; $P \leftarrow \{\boldsymbol{c}\}$;
for *each head* $h' \neq h$ **do**
  $\boldsymbol{c}' \leftarrow \arg\min_{\boldsymbol{c}' \in M_q^{l,h'}} (S_q^{l,h'})$;
  $S_q^{l,h'} \leftarrow S_q^{l,h'} \setminus \{s_{\boldsymbol{c}'}\}$; $P \leftarrow P \cup \{\boldsymbol{c}'\}$;

Algorithm 2: Attention-Level Consistency.

Input: Selected channel $\boldsymbol{c} \in M_q^{l,h}$, score sets $\{S_q^{l,h}, S_k^{l,h}\}$
Output: Expanded channels P for sparsity

$S_q^{l,h} \leftarrow S_q^{l,h} \setminus \{s_{\boldsymbol{c}}\}$; $P \leftarrow \{\boldsymbol{c}\}$
$i = index(M_q^{l,h}, \boldsymbol{c})$; $\boldsymbol{c}' \leftarrow M_k^{l,h}[i]$
$S_k^{l,h} \leftarrow S_k^{l,h} \setminus \{s_{\boldsymbol{c}'}\}$; $P \leftarrow P \cup \{\boldsymbol{c}'\}$;

Algorithm 3: Channel Pruning With Consistencies.

Input: Global importance score set S, target FLOPs reduction ratio r_{target}
Output: Channels P for sparsity, current FLOPs reduction ratio $r_{current}$

while $r_{current} < r_{target}$ **do**
  $\boldsymbol{c} \leftarrow \arg\min_c S$;
  if $\boldsymbol{c} \in M_{proj}^l \cup M_{fc1}^l$ **then**
    $P \leftarrow P \cup \{\boldsymbol{c}\}$; $S \leftarrow S \setminus \{s_{\boldsymbol{c}}\}$;
  else if $\boldsymbol{c} \in M_v^{l,h}$ **then**
    $P_{head} \leftarrow$ Head-level consistency($\boldsymbol{c}$);
    $P \leftarrow P \cup P_{head}$; $S \leftarrow S \setminus S_{P_{head}}$;
  else if $\boldsymbol{c} \in M_q^{l,h} \cup M_k^{l,h}$ **then**
    $P_{head} \leftarrow$ Head-level consistency($\boldsymbol{c}$);
    for *each channel* $\boldsymbol{c}' \in P_{head}$ **do**
      $P_{attn} \leftarrow$ Attention-level consistency($\boldsymbol{c}'$);
      $P \leftarrow P \cup P_{attn}$; $S \leftarrow S \setminus S_{P_{attn}}$;
  Update $r_{current}$

importance, thus being pruned or maintained consistently. This greatly benefit error-free interaction for the attention.

Pipeline: Algortihm 3 illustrates the proposed consistent dynamic channel pruning process. During finetuning, unconstrained channels, *i.e.*, $\boldsymbol{c} \in M_{proj}^l \cup M_{fc1}^l$, are directly added to P. For channel $\boldsymbol{c} \in M_q^{l,h} \cup M_k^{l,h}$, we apply the proposed head-level consistency to it. After that, we employ the attention-level consistency to the newly to-be-sparse channels. For $\boldsymbol{c} \in M_v^{l,h}$, we only apply head-level consistency. After finetuning, channels of zero importance in M are pruned, resulting in the pruned compactor $\overline{M}$. $\overline{M}$ can thus be merged with its preceding weight W to obtain the compressed model.

IV. Experiments

We first compare our method with state-of-the-arts on ImageNet [32] to verify the high performance and fast inference speed obtained by our method (Section IV-A), following [23], [24], [28]. We then evaluate the performance of our accelerated backbones on various downstream pixel-level vision tasks to demonstrate their strong transferability (Section IV-B). Additionally, we investigate impacts of each component by comprehensive analyses on ImageNet, following [28] (Section IV-C). Furthermore, we provide more evaluation, analyses, and visualization in Section IV-D. The float operations (FLOPs) of models are measured by fvcore [75] and the throughput is evaluated on a single NVIDIA RTX-3090 GPU with a batch size of 256, by default. For compared methods, we utilize their published pretrained models to obtain throughputs.

TABLE II
Hyper-Parameters During Fine-Tuning

hyper-parameter	value
optimizer	AdamW
base learning rate	1e-4
weight decay	0.05
optimizer momentum	β_1 = 0.99 (compactor), 0.9 (other) β_2 = 0.999
batch size	256 (Tiny/Small), 128 (Base)
learning rate schedule	cosine decay
label smoothing	0.1
mixup	0.8
cutmix	1.0
distillation loss	cross entropy loss
distillation-alpha	0.1 (Tiny), 0.25 (Small/Base)

A. Comparison Results on ImageNet

1) Implementation Details: We evaluate our proposed method on three different sizes of DeiT [6], *i.e.*, DeiT-Tiny, DeiT-Small, and DeiT-Base. Our experiments are deployed with Pytorch [77] on RTX-3090 GPUs. In CDCP, following [70], r_{target} is initialized to zero, which is then increased by 0.025% every 25 iterations until achieving the given reduction ratio. Meanwhile, we re-construct P every same iterations. Besides, we start to increase r_{target} and re-construct P after 30 epochs. λ in (2) is empirically set to 1e-5. Table II reports the detailed hyper-parameters during training, most of which are the same as [28]. Following [23], [24], [28], the corresponding original models are used for hard distillation.

2) Results: As shown in Table III, our proposed method can consistently outperform previous methods across all three models, as evidenced by superior performance in terms of the Top-1 accuracy, the FLOPs reduction ratio, and the inference speed. Specifically, under similar FLOPs reduction ratios, our method outperforms the state-of-the-art VTC-LFC [28] by 0.7% and 0.4% in terms of Top-1 accuracy on DeiT-Tiny and DeiT-Small, respectively, all while achieving significantly faster inference speeds. Such improvements can be attributed to the effective token merging and channel pruning by our ATME and CDCP, respectively. Compared with methods that obtain comparable accuracy to ours, such as dTPS [35] and SPViT [43], our method can achieve much higher FLOPs reductions. We can see that in

TABLE III
COMPARISON WITH STATE-OF-THE-ARTS ON IMAGENET

Model	Param. (M)	FLOPs (G)	Speed (↑)	Top-1 (%)
DeiT-Tiny	5.7	1.3	1.0×	72.2
S^2ViTE [21]	4.2	1.0	-	70.1
SPViT [43]	4.9	1.0	1.1×	70.7
ToMe [34]	5.7	0.7	1.5×	71.3
UVC [23]	-	0.6	-	71.3
SAViT [24]	4.2	0.9	-	70.7
VTC-LFC [28]	5.1	0.7	0.9×	71.6
CAIT (ours)	5.1	**0.6**	**1.7**×	**72.3**
DeiT-Small	22.1	4.6	1.0×	79.8
CP-ViT [76]	22.1	2.7	-	79.1
EViT [30]	22.1	2.3	1.8×	78.5
IA-RED^2 [33]	22.1	3.2	1.5×	79.1
dTPS [35]	22.8	3.0	1.5×	80.1
S^2ViTE [21]	14.6	3.2	-	79.2
SPViT [43]	16.4	3.3	1.0×	78.3
ToMe [34]	22.1	2.7	1.5×	79.4
UVC [23]	-	2.3	-	78.8
SAViT [24]	14.7	3.1	-	80.1
VTC-LFC [28]	17.7	2.1	1.1×	79.8
CAIT (ours)	18.4	**2.1**	**1.9**×	**80.2**
DeiT-Base	86.4	17.6	1.0×	81.8
EViT [30]	86.4	11.6	1.5×	81.3
IA-RED^2 [33]	86.4	11.8	1.4×	80.3
S^2ViTE [21]	56.8	11.8	-	82.2
SPViT [43]	62.3	11.7	1.0×	81.6
UVC [23]	-	8.0	-	80.6
VTC-LFC [28]	63.5	7.5	1.2×	81.3
CAIT (ours)	71.3	**7.4**	**2.1**×	**81.6**

the proposed method, the fruitful FLOPs reduction can be sufficiently transformed into the significant inference acceleration. Notably, our compressed DeiT-Tiny, DeiT-Small, and DeiT-Base models can achieve 1.7×, 1.9×, and 2.1× inference speedups, respectively, while enjoying no or little accuracy drops. These results well demonstrate the effectiveness and the superiority of our method.

B. Transferability on Downstream Pixel-Level Tasks

1) Results on Semantic Segmentation: Most existing token pruning methods generally reduce the number of tokens in an unstructured manner [22], [25], [28], [30], [33], [34], [35], *i.e.*, by dropping tokens sparsely, which however inevitably disrupts the complete spatial structure of images. Therefore, the accelerated ViTs are not suitable for downstream pixel-level vision tasks, like semantic segmentation. To verify the impact of unstructured token pruning on downstream vision tasks, we conduct experiments with the start-of-the-art VTC-LFC [28] on the ADE20k [79] dataset. Additionally, in contrast, our proposed method can preserve the spatial integrity and effectively adapt to downstream tasks that need a complete spatial structure of images. Therefore, we also conduct experiments on the ADE20 k dataset to verify such a transferability. We introduce state-of-the-art Evo-ViT [41] as one baseline, which can also maintain spatial structure of input images as ours.

Following [49], [80], we integrate accelerated backbones into three advanced segmentation methods, *i.e.*, Semantic FPN [78], UperNet [36], and Mask2Former [37]. We train for 80 k, 160 k, and 160 k iterations for these three segmentation methods, respectively. Besides, we adopt the AdamW [81] optimizer with the learning rate of 6e-5 and weight decay of 0.01, as in [5]. The input resolution is set to 512×512 and all models are trained using batch size of 32. We report the performance with standard single scale protocol as in [49], [80]. Additionally, the encoder speedup (En. sp.) and overall speedup (Over. sp.) are evaluated on a single RTX-3090 GPU with a batch size of 32, where the encoder contains the backbone, upsampling and downsampling modules. Our implementation is based on mmsegmentation [82].

As VTC-LFC [28] produces sparse feature maps, following [83], [84], we use mask tokens to fill the dropped positions before feeding them into the semantic segmentation decoder, which is denoted as "VTC-LFC-unstructured". As shown in Table IV, due to the impaired spatial integrity of feature maps, CNN-based decoders, *i.e.*, Semantic FPN [78] and UperNet [36], result in poor results. It is consistent with observations in previous works [84], [85] that CNNs exhibit significantly worse performance when dealing with sparse feature maps, which can be attributed to the disrupted data distribution of pixel values and vanished patterns of visual representations. Besides, with the Transformer-based decoder, *i.e.*, Mask2Former [37], "VTC-LFC-unstructured" demonstrates a significant inferiority to DeiT-Small, with a considerable margin of 2.0% mIoU. These results well show the harmful impacts caused by unstructured token pruning when transferring the accelerated model to downstream structured vision task of semantic segmentation. Furthermore, we propose to record the dropped tokens and then use them to fill the corresponding positions when constructing feature maps to be fed into the segmentation decoder, thus ensuring the spatial integrity of patches, which is denoted as "VTC-LFC-structured". As shown in Table IV, reasonably, "VTC-LFC-structured" outperforms "VTC-LFC-unstructured" across three segmentation methods.

Furthermore, as shown in Table IV, our method not only exhibits superior performance but also boasts fast inference speed across all semantic segmentation methods. Specifically, our ATME yields impressive overall speedups of 1.27×, 1.23×, and 1.18×, respectively, across three distinct segmentation methods, while maintaining optimal performance. Our ATME outperforms "VTC-LFC-structured" by great margins of 1%, 1.1%, 1.3% mIoUs on all segmentation decoders, respectively, with notably faster inference speedup. Besides, our ATME significantly outperforms Evo-ViT [41] with 1.8%, 2.3%, and 1.7% mIoU on three segmentation heads, respectively. It well indicates the superiority of asymmetric token merging in preserving spatial integrity, compared with Evo-ViT that can potentially harm token features. Besides, on top of ATME, our CAIT can further enhance the overall inference speed. These results well demonstrate the remarkable adaptability of the proposed method to downstream vision tasks.

2) Results on Object Detection & Instance Segment.: To verify the strong transferability of our method on various downstream pixel-level vision tasks, we experiment over COCO-2017 [86] to evaluate the performance on object detection and instance segmentation. Following [87], we integrate the

TABLE IV
RESULTS ON SEMANTIC SEGMENTATION

	Semantic FPN [78]			UperNet [36]			Mask2Former [37]		
Backbone	mIoU	En. sp. ↑	Over. sp. ↑	mIoU	En. sp. ↑	Over. sp. ↑	mIoU	En. sp. ↑	Over. sp. ↑
DeiT-Small	44.3	1.00×	1.00×	44.9	1.00×	1.00×	47.2	1.00×	1.00×
VTC-LFC-unstructured [28]	0.1	1.01×	1.01×	0.1	1.01×	1.01×	45.2	1.16×	1.05×
VTC-LFC-structured [28]	43.5	1.01×	1.00×	44.2	1.00×	1.00×	46.3	1.15×	1.05×
Evo-ViT [41]	42.7	1.34×	1.20×	43.0	1.33×	1.18×	45.9	1.41×	1.14×
ATME	**44.5**	**1.46×**	**1.27×**	**45.3**	**1.43×**	**1.23×**	**47.6**	**1.54×**	**1.18×**
CAIT	**44.5**	**1.52×**	**1.31×**	**45.6**	**1.48×**	**1.26×**	**47.2**	**1.64×**	**1.21×**

TABLE V
RESULTS ON OBJECT DETECTION & INSTANCE SEGMENT

Method	AP^{box}	AP^{mask}	Speed ↑
DeiT-Small	38.4	35.2	1.00×
VTC-LFC-structured	37.5	34.4	1.01×
ATME	**38.7**	**35.2**	**1.46×**
CAIT	**38.8**	**35.6**	**1.52×**

TABLE VI
RESULTS ON MEDICAL IMAGE SEGMENTATION

Method	Synapse	ACDC	Polyp	Speed ↑
DeiT-Small	76.5	88.6	78.9	1.00×
VTC-LFC-structured	74.1	87.6	65.9	1.01×
CAIT	**76.4**	**88.4**	**78.9**	**1.52×**

TABLE VII
RESULTS ON AERIAL SEMANTIC SEGMENTATION

Method	Potsdam		iSAID	Speed ↑
	OA	mF1	mIoU	
DeiT-Small	88.6	90.9	60.8	1.00×
VTC-LFC-structured	88.2	90.6	58.7	1.01×
CAIT	**88.5**	**90.9**	**61.1**	**1.52×**

TABLE VIII
ABLATION STUDY ON DEIT-TINY AND DEIT-SMALL

Method	DeiT-Tiny			DeiT-Small		
	Top-1	Param.	FLOPs ↓	Top-1	Param.	FLOPs ↓
Original	72.2%	5.7M	-	79.8%	22.1M	-
Original-600e	73.5%	5.7M	-	81.0%	22.1M	-
ATME	71.9%	5.9M	50.2%	79.8%	22.9M	54.4%
CDCP	71.9%	4.2M	31.2%	79.8%	13.9M	38.1%
CAIT	**72.3%**	**5.1M**	**50.5%**	**80.2%**	**18.4M**	**54.4%**

accelerated backbones into Mask-RCNN [38]. We adopt the AdamW optimizer with a initial learning rate of 2×10^{-4}. The models are trained for 12 epochs with the input resolution of 1333×800. We introduce the state-of-the-art "VTC-LFC-structured" as the baseline method.

As shown in Table V, thanks to preserving the complete spatial structure of image patches, our ATME significantly outperforms "VTC-LFC-structured" by 1.2 AP^{box} and 0.8 AP^{mask}, respectively. Besides, our CAIT achieves 1.52× speedup without performance degradation, well demonstrating the strong transferability of our method.

3) Results on Medical and Aerial Segmentation: In order to demonstrate the generalizability of our accelerated backbones to out-of-domain downstream tasks, we conduct experiments on medical image segmentation and aerial semantic segmentation tasks. Specifically, for medical image segmentation, following [88], we integrate accelerated backbones into the CASCADE framework [88] and evaluate the performance on widely used Synapse multi-organ dataset [89], ACDC dataset [90] and Polyp datasets [91], [92]. We follow [88] to report DICE score for all datasets. Regarding to aerial semantic segmentation, we follow [93] to adopt UperNet [36] as the unified segmentation framework. Besides, we conduct experiments on ISPRS Potsdam dataset [94] and the large-scale segmentation benchmark, *i.e.*, iSAID [95]. We follow [93] to report the overall accuracy (OA) and mean F1 score (mF1) for the Potsdam dataset, and mIoU for the iSAID dataset. We introduce state-of-the-art "VTC-LFC-structured" as baseline.

As shown in Table VI, benefiting from the preserved spatial integrity, our CAIT significantly outperforms "VTC-LFC-structured" by 2.3, 0.8 and 13.0 DICE score on Synapse, ACDC and Polyp datasets, respectively. Meanwhile, compared with DeiT-Small, it demonstrates 1.52× speedup with comparable performance across all three datasets. Similarly, as shown in Table VII, our CAIT significantly surpasses "VTC-LFC-structured" with 0.3 mF1 and 2.4 mIoU on Potsdam and iSAID, respectively. Besides, it attains a notable speed improvement of 1.52× without compromise in performance. Overall, thanks to well-preserved spatial integrity and dynamic fine-grained compression optimization by ATME and CDCP, respectively, our CAIT demonstrates robust generalizability across out-of-domain downstream tasks.

Remark: Extensive experiments on ImageNet and various downstream pixel-level tasks well demonstrate our superiority in terms of accuracy, efficiency and transferability. Guided by triple-win compression principles, our method successfully deliver accelerated models with high accuracy, fast inference speed, and favorable transferability all at once, showing promising performance in practical scenarios.

C. Model Analyses

1) Ablation Study: We conduct experiments with DeiT-Tiny and DeiT-Small, following [28], [35]. As shown in Table VIII,

TABLE IX
COMPARISON WITH ALTERNATIVE METHODS ON DEIT-SMALL (TOP-1: 79.8%)

Token Pruning	Channel Pruning	Top-1 (%)	FLOPs (↓%)	Speed ↑
EViT	-	79.6	43.3%	1.7×
LFE	-	80.1	43.3%	1.5×
ATME	**-**	**80.0**	**43.3%**	**1.8×**
-	NViT	78.9	32.8%	1.2×
-	LFS	79.4	32.8%	1.2×
-	**CDCP**	**79.8**	**32.8%**	**1.2×**
LFE	LFS	79.1	55.0%	1.6×
LFE	CDCP	79.6	55.0%	1.6×
ATME	LFS	79.1	55.0%	2.0×
ATME	**CDCP**	**79.5**	**55.0%**	**2.0×**

compared with original models, our ATME can obtain comparable accuracy while reducing 50.2% and 54.4% FLOPs for DeiT-Tiny and DeiT-Small, respectively. The proposed CDCP can obtain sufficient FLOPS reduction as well. These results can demonstrate the effectiveness of ATME and CDCP. We can also observe that, compared with CDCP, our ATME, as a token pruning method, can obtain superior performance. This result is consistent with observations in prior works [27], [28], [30] that for ViT models, compressing tokens can achieve more outcomes than compressing channels. Therefore, in practice, we follow [28] to assign more FLOPs reduction ratio on ATME. Specifically, given a desired ratio of overall FLOPs reduction, we first prioritize the strategy of uniformly dividing layers for token pruning, and then adjust the FLOPs reduction ratio of channel pruning to exactly match the target overall FLOPs reduction. Furthermore, it can be observed that compared with ATME and CDCP, the final model, CAIT, achieves the best performance. This is attributed to that CAIT can simultaneously eliminate the data level redundancy by ATME and model level redundancy by CDCP, achieving optimal outcomes. Besides, we also continue training the pretrained DeiT-Tiny and DeiT-Small for 300 epochs under the same setting, denoted as "Original-600e", which leads to the same epochs as compressing pretrained models and serves as the performance upper bounds. Compared with them, we note that our CAIT also shows competitive performance after significant computation reduction. These results demonstrate the effectiveness and superiority of CAIT.

2) Superiority to Alternative Methods: To verify the superiority of our proposed ATME and CDCP over existing token pruning and channel pruning methods, we conduct experiments on ImageNet with only compressing tokens, channels, and both. Following [28], we introduce two state-of-the-art token pruning methods, *i.e.*, EViT [30] and LFE [28], and two state-of-the-art channel pruning methods, *i.e.*, NViT [27] and LFS [28], on DeiT-Small, as baselines for ATME and CDCP, respectively. When compressing both tokens and channels, we select better token pruning baseline method, *i.e.*, LFE [28] and better channel pruning baseline method, *i.e.*, LFS [28] for combinations. Results of compared baselines are borrowed from [28] directly. For fair comparison, we employ the same training setting as [28].

As shown in Table IX, our ATME outperforms EViT by 0.4% Top-1 accuracy under the same FLOPs reduction. Compared with LFE, our ATME achieves significantly faster inference speed while obtaining comparable accuracy. For channel pruning, our CDCP can outperform NViT and LFS by 0.9% and 0.4% accuracy gains, respectively. When compressing both tokens and channels, our joint compression method is still superior to other combinations, *i.e.*, ATME+LFS, LFE+CDCP, and LFE+LFS. Overall, our ATME and CDCP show their effectiveness compared with other token pruning and channel pruning methods, respectively.

TABLE X
IMPACT OF ASYMMETRY IN ATME WITH DEIT-TINY (TOP-1: 72.2%)

Method	Top-1 (%)	Param. (M)	FLOPs (↓%)	Speed ↑
symmetry	71.2	6.0	50.7%	1.9×
HTM	71.5	5.9	49.6%	1.8×
VTM	71.5	5.9	49.6%	1.8×
diag	71.2	5.9	49.2%	1.8×
ATME	**71.9**	5.9	**50.2%**	**1.9×**

TABLE XI
IMPACT OF CONSISTENCIES IN CDCP WITH DEIT-TINY (TOP-1: 72.2%)

Method	Top-1 (%)	Param. (M)	FLOPs (↓%)	Speed ↑
S^2ViTE [21]	70.1	4.2	23.7%	1.1×
w/o both	71.0	4.3	25.1%	1.2×
w/o head	71.3	4.4	25.1%	1.2×
w/o attn	72.1	4.4	25.1%	1.2×
CDCP	**72.7**	4.5	**25.1%**	**1.2×**

3) Asymmetry in ATME: We investigate the beneficial impacts of asymmetry in ATME. We introduce four baselines:

1) simultaneously using horizontal and vertical token merging as one operation, denoted as "symmetry", in which we group and concatenate four adjacent tokens in both horizontal and vertical directions, *i.e.*, in a 2×2 patch;
2) only using horizontal token merging;
3) only using vertical token merging.
4) diagonally merging tokens, denoted as "diag".

As shown in Table X, ATME can obtain better performance. Specifically, compared with "symmetry", ATME progressively reduces the number of tokens in a moderate way, forbidding drastic losses of token information, thus achieving a 0.7% accuracy gain. Compared with HTM and VTM, ATME can maintain a more regular spatial structure for patches, resulting in a 0.4% performance improvement. Compared with "diag", ATME can enjoy more locality inductive bias and obtain the improvement of 0.7% accuracy. These results show favorable advantage of asymmetry in ATME.

4) Consistencies in CDCP: We verify the positive effects of head-level consistency and attention-level consistency used in CDCP. Additionally, we introduce S^2 ViTE [21] as the baseline method, because it is a remarkable state-of-the-art dynamic channel pruning method. As shown in Table XI, head-level and attention-level consistencies can consistently achieve performance improvements. Specifically, head-level consistency leads to a 1.4% (CDCP 72.7% vs "w/o head" 71.3%) accuracy gain. Attention-level consistency obtains a 0.6% (CDCP 72.7% vs "w/o attn" 72.1%) performance improvement. Besides, our CDCP significantly outperforms the baseline "w/o both" and

TABLE XII
RESULTS ON LV-VIT AND SWIN TRANSFORMER

Method	Top-1 (%)	FLOPs (↓%)	Speed ↑
LV-ViT-S [80]	83.2	-	1.0×
NViT [27]+EViT [30]	81.5	49.2%	1.8×
VTC-LFC [28]	81.8	50.8%	1.2×
CAIT	**82.2**	**53.8%**	**1.9×**
Swin-Tiny [5]	81.1	-	1.0×
SPViT [43]	80.1	24.4%	-
VTC-LFC [28]	80.3	26.7%	1.2×
CAIT	**80.6**	**26.7%**	**1.2×**

TABLE XIII
IMPACT OF DIFFERENT USAGES OF ATME

Method	Top-1 (%)	Param. (M)	FLOPs (G)	Speed ↑
DeiT-Tiny	72.2	5.7	1.3	1.0×
DeiT-half depth	64.0	3.0	0.6	1.9×
DeiT-half dim	66.3	2.8	0.6	1.2×
ATME-scratch	71.1	5.9	0.6	1.9×
ATME+Conv	71.9	6.0	0.6	1.9×
ATME	**71.9**	5.9	0.6	**1.9×**

TABLE XIV
ATME IN HIERARCHICAL ARCHITECTURES

Method	Top-1 (%)	Param. (M)	FLOPs (↓%)	Speed ↑
Swin-Tiny [5]	81.1	28.3	-	1.0×
CDCP	80.6	24.6	26.7	1.2×
CAIT	**80.5**	25.5	**35.4**	**1.3×**

S ^{2} ViTE [21]. Such improvements can be attributed to the fine-grained compression with head-level and attention-level consistencies for ViTs.

5) Compression on Other ViT Models: To explore the performance of our method on other variants of ViTs, we conduct experiments on LV-ViT [80] and Swin Transformer [5]. Following [28], we adopt ATME and CDCP on LV-ViT, and employ CDCP to Swin. Meanwhile, for simplicity, the proposed token labels in the original LV-ViT are not used during training, like [28]. As shown in Table XII, our method can consistently achieve the state-of-the-art performance on both models. Specifically, on LV-ViT, our method outperforms VTC-LFC [28] with 0.4% higher accuracy while achieving significantly faster acceleration (CAIT 1.9× vs VTC-LFC 1.2×). For Swin Transformer, our compressed model can also obtain accuracy gains of 0.5% and 0.3% compared with SPViT [43]/VTC-LFC [28], respectively, under the similar FLOPs reduction ratio. These results well demonstrate the generalization of our method on other ViT variants. Besides, we can observe that LV-ViT and Swin Transformer generally suffer more accuracy drop after pruning than DeiT, which is consistent with previous works [28]. We hypothesize the reason lies in the architectural differences among LV-ViT, Swin Transformer, and DeiT. Specifically, LV-ViT adopts a narrower expansion ratio in FFN and a deeper layout to improve efficiency. It also leverages token labeling to introduce individual location-specific supervision. Swin Transformer adopts the hierarchical structure and shifted window design to enhance efficiency. Therefore, LV-ViT and Swin Transformer exhibit less data-level (*i.e.*, tokens) redundancy and model-level (*i.e.*, parameters) redundancy, compared with DeiT. They thus suffer more accuracy drop after pruning. Additionally, our proposed method can consistently outperform existing methods on LV-ViT and Swin Transformer, showing promising performance for pruning various ViTs.

D. Discussion

1) Locality Matters for ATME: We provide more insightful analyses for ATME. The proposed ATME uniformly aggregates features of neighboring tokens, which can be regarded as a general architecture for modern ViTs. Thus, we construct a vision transformer model whose architecture is the same as our ATME. Then, we train this model for 600 epochs from scratch with hard distillation of the pretrained model. We denote this model as "ATME-scratch". Besides, we introduce two additional models whose FLOPs are similar to ATME-scratch's. One involves halving the depth of DeiT-Tiny, which reduces the number of blocks. The other involves halving the width of DeiT-Tiny, which reduces the embedding dimension. We denote these two models as "DeiT-half depth" and "DeiT-half dim", respectively, which are trained under the same setting as "ATME-scratch". We compare these three models with the one obtained by our ATME pruning method. As shown in Table XIII, "ATME-scratch" significantly outperforms "DeiT-half depth" and "DeiT-half dim" by 7.1% and 4.8% in terms of Top-1 accuracy, respectively. This may be attributed to the inductive bias of locality introduced by our ATME strategy. We also incorporate the extra convolution into the HTM and VTM by appending the depthwise convolution with kernel size of 3×3, which is denoted as "ATME+Conv". Compared with ATME, it brings negligible performance gain due to the inherent locality in ATME. Moreover, "ATME-scratch" is inferior to DeiT-Tiny by a great margin of 1.1% accuracy. In contrast, our ATME can result in only a 0.3% drop, compared with vanilla DeiT-Tiny, while achieving a superior speedup of 1.9×. It indicates that in addition to introducing the locality, our ATME can further preserve the pretrained model's ability to capture visual features and prevent knowledge forgetting during pruning. Thanks to them, our ATME can well serve as a compression methodology for ViTs, delivering high performance and fast inference speed.

2) ATME in Hierarchical Architectures: As an efficient token pruning strategy for DeiT, our proposed ATME can also transfer to hierarchical architectures, *e.g.*, Swin Transformer. We conduct experiments under the same setting in Section IV-C5 to verify this. Specifically, we adopt HTM and VTM at the last two layers of the 3-th Stage in Swin-Tiny, respectively, which results in a FLOPs reduction of 20.1%. We further perform channel pruning on it, leading to an overall 35.4% FLOPs reduction. As shown in Table XIV, CAIT obtains a comparable accuracy with only performing channel pruning on Swin-Tiny, but with a much larger FLOPs reduction (35.4% vs. 26.7%) and a more significant inference speedup (1.3x vs. 1.2x). It well

TABLE XV
COMPARISON WITH TOME AND DIFFRATE ON IMAGENET. * INDICATES THAT TOME IS ADOPTED IN THE SAME PRUNING LAYERS AS OUR ATME.

Method	Top-1 (%)	Param. (M)	FLOPs (↓%)	Speed ↑
DeiT-Small	79.8	22.1	-	1.0×
ToMe	79.9	22.1	41.3	1.5×
ToMe+Param	80.0	22.6	42.7	1.5×
ToMe*	80.0	22.1	41.6	1.7×
DiffRate	80.1	22.1	41.3	1.6×
ATME	**80.0**	22.6	**43.3**	**1.8**×

TABLE XVI
COMPARISON WITH TOME AND DIFFRATE ON ADE20 K USING SEMANTIC FPN. * INDICATES THAT TOME IS ADOPTED IN THE SAME PRUNING LAYERS AS OUR ATME.

Method	mIoU	Param. (M)	FLOPs (↓%)	Speed ↑
DeiT-Small	44.3	22.1	-	1.0×
ToMe	44.0	22.1	41.3	1.5×
ToMe+Param	44.0	22.6	42.7	1.5×
ToMe*	43.9	22.1	41.6	1.7×
DiffRate	43.1	22.1	41.6	1.6×
ATME	**44.5**	22.6	**43.3**	**1.8**×

TABLE XVII
COMPARISON WITH STVIT-R ON IMAGENET. † MEANS THE REPRODUCED PERFORMANCE USING THE OFFICIAL CODE.

Method	Top-1 (%)	Param.(M)	FLOPs (↓%)	Speed ↑
Swin-Tiny	81.3	28.3	-	1.0×
STViT-R	80.5†	28.3	19.6	1.2×
ATME	**81.1**	28.9	**20.1**	**1.2**×
Swin-Small	83.2	49.6	-	1.0×
STViT-R	82.5†/82.7	49.6	33.0	1.3×
ATME	**83.0**	50.2	**33.7**	**1.3**×

demonstrates the effectiveness of our token compression method in transferring to hierarchical architectures.

3) Superiority of ATME to Others: ToMe [34] and DiffRate [96] are existing state-of-the-art token pruning methods, which leverage bipartite soft matching to merge similar tokens. To demonstrate the superiority of our proposed ATME for token pruning, we compare our strategy with three variants:

1) using the strategy in ToMe in the same pruning layers as our ATME; and
2) performing ToMe at every layer as in their paper [34]; and
3) adopting the DiffRate for token pruning.

Besides, we also introduce extra parameters to ToMe like ours. To maintain the similar number of parameters and the FLOPs reduction, we incorporate the Linear layer after the merging of ToMe in every three blocks and slightly increase the number of merged tokens, which is denoted as "ToMe+Param".

We first conduct experiments on ImageNet under the same experimental setups to investigate their performance based on DeiT-Small. Specifically, we finetune ToMe and DiffRate under the same training setting as ours. As shown in Table XV, our ATME obtains a comparable accuracy with the ToMe variants and DiffRate under a larger FLOPs reduction, demonstrating the effectiveness of our asymmetric token merging method. Besides, the strategy in ToMe and DiffRate employs the complex bipartite similarity matching with complex operators, while our ATME simply merges neighboring tokens and utilizes fast tensor manipulations. Our ATME thus affords a significant advantage for various devices and platforms, particularly those with limited computation ability or lacking support for complex operators. As evidenced in Table XV, our ATME is more friendly to latency and leads to an advantageous inference speedup compared with ToMe and DiffRate.

More importantly, the strategy in ToMe and DiffRate merges tokens sparsely at each pruning layer, resulting in the disruption of the spatial integrity of images and restricting the transferability of compressed models to downstream structured vision tasks. In contrast, our ATME can well preserve the complete structure of patches and maintain the strong transferability of ViTs. We further conduct experiments on the downstream semantic segmentation task to verify this, by Semantic FPN segmentation method. To transfer the strategy in ToMe and DiffRate to the downstream task, we track which tokens get merged and then unmerge tokens, *i.e.*, using the merged token to fill the corresponding empty positions, when constructing feature maps to be fed into the segmentation decoder. Besides, for DiffRate which also sparsely drops tokens, we adopt the same strategy as "VTC-LFC-structured" to obtain the complete feature map. As shown in Table XVI, since DiffRate's pruning strategy is specifically tuned on ImageNet, its performance significantly degrades when transferred to the downstream task. Besides, our ATME outperforms others with considerable margins, along with a larger inference speedup, showing the favorable transferability for downstream structured vision tasks.

We further compare our ATME with STViT-R [42], which presents the recovery module and dumbbell unit to perform token pruning and adapt to downstream tasks. Note that its another variant STViT hinders the application for downstream tasks [42] due to the side effect of losing nearly all the detailed information, we thus leave out it. We follow STViT-R to leverage ATME on Swin Transformer and train from scratch for 300 epochs. To achieve comparable FLOPs reduction with STViT-R, we adopt the HTM and VTM at the 4-th and 5-th layers of the third stage for Swin-Tiny, and at the 6-th and 12-th layers of the third Stage for Swin-Small, respectively. As shown in Table XVII, on ImageNet, our ATME significantly outperforms STViT-R by 0.6% and 0.5% top-1 accuracies on Swin-Tiny and Swin-Small, respectively. It shows the favorable high accuracy of ATME. Besides, we follow STViT-R to transfer the models to object detection and instance segmentation using Cascade Mask R-CNN, and semantic segmentation using UperNet. All the training settings follow STViT-R. As shown in Table XVIII, our ATME surpasses STViT-R by 0.7 AP^{box} and 0.6 AP^{mask} on Swin-Tiny. Additionally, ATME outperforms STViT-R by considerable margins on semantic segmentation. Due to that STViT-R only remains the tokens with high-level semantic information, it loses nearly all the detailed pixel-level information and thus suffers inferior performance on downstream dense vision tasks. In contrast, our ATME can well maintain the complete spatial structure and dense position information, showing strong transferability. Furthermore, we also compare ATME

TABLE XVIII
COMPARISON WITH STVIT-R ON OBJECT DETECTION (AP^{box}) AND INSTANCE SEGMENTATION (AP^{mask}) ON COCO, AND SEMANTIC SEGMENTATION (MIOU) ON ADE20K

Method	AP^{box}	AP^{mask}	mIoU	Speed ↑
Swin-Tiny	50.5	43.7	45.8	1.0×
STViT-R	49.4†	42.7†	43.9†	1.2×
ATME	**50.1**	**43.3**	**45.2**	**1.2×**
Swin-Small	51.8	44.7	49.5	1.0×
STViT-R	51.6†/51.8	44.7†/44.7	46.4†/48.3	1.3×
ATME	**51.8**	**44.7**	**48.5**	**1.3×**

TABLE XIX
COMPARISON WITH STVIT-R ON MEDICAL IMAGE SEGMENTATION

Method	Synapse	ACDC	Polyp	Speed ↑
Swin-Tiny	79.5	90.0	80.6	1.0×
STViT-R	79.0	89.7	79.8	1.2×
ATME	**79.3**	**90.1**	**80.5**	**1.2×**

TABLE XX
COMPARISON WITH STVIT-R ON AERIAL SEMANTIC SEGMENTATION

Method	Potsdam		iSAID	Speed ↑
	OA	mF1	mIoU	
Swin-Tiny	91.2	90.6	64.6	1.0×
STViT-R	90.9	90.4	63.2	1.2×
ATME	**91.1**	**90.6**	**64.6**	**1.2×**

TABLE XXI
RESULTS FOR DIFFERENT EPOCHS FOR STARTING INCREASING r_{target} ON DEIT-TINY

warmup epochs	Top-1 (%)	FLOPs (↓%)	Speed ↑
0	72.26	50.5	1.7×
15	72.31	50.5	1.7×
30	72.34	50.5	1.7×

and STViT-R on out-of-domain downstream tasks, *i.e.*, medical image segmentation and aerial semantic segmentation. We adopt the same experimental setups as Section IV-B3. As shown in Table XIX, our ATME outperforms STViT-R by 0.7 DICE score on Polyp dataset. As shown in Table XX, ATME obtains 1.4 mIoU improvement over STViT-R on iSAID. These results demonstrate robust generalizability of ATME over STViT-R on downstream vision tasks.

4) Robustness of CDCP to Compression Schedule: We follow [70] to set the compression training schedule of CDCP. To verify that the pruning performance of our CDCP is not sensitive to different compression schedules, we conduct experiments on DeiT-Tiny to analyze the effects of the epoch for starting increasing r_{target} and interval iterations for the reconstruction of P. As shown in Table XXI and Table XXII, we can see that they do not make significant differences, indicating the robustness of CDCP. Therefore, the effectiveness of our method is general and not limited by specific schedules.

TABLE XXII
RESULTS FOR DIFFERENT INTERVAL ITERATIONS FOR THE RECONSTRUCTION OF P ON DEIT-TINY

interval iterations	Top-1 (%)	FLOPs (↓%)	Speed ↑
25	72.34	50.5	1.7×
50	72.37	50.5	1.7×
75	72.46	50.5	1.7×

TABLE XXIII
RESULTS FOR DIFFERENT FINE-TUNING EPOCHS ON DEIT-SMALL. THE SUFFIX "-XE" MEANS X EPOCHS

Method	Top-1 (%)	FLOPs (↓%)	Speed ↑
DeiT-Small	79.8	-	1.0×
CAIT-30e	76.7	55.0	2.0×
CAIT-150e	79.5	55.0	2.0×
CAIT-300e	80.2	55.0	2.0×

5) Different Fine-Tuning Epochs for CAIT: To investigate the performance of different fine-tuning epochs of our CAIT, we conduct experiments on DeiT-Small. As shown in Table XXIII, due to introducing parameters for extra optimization, our method benefits more from longer fine-tuning epochs, and results in a high performance upper bound. Specifically, our CAIT-150e and CAIT-300e enjoys a 2.0× inference speedup with no or little accuracy drops.

6) Parameter Distribution of Pruned Models: We visualize the parameter distribution of pruned DeiT-Tiny, DeiT-Small and DeiT-Base. Fig. 6 presents the saved ratios of channels in each block. It reveals that middle and deep blocks tend to retain more channels than shallow blocks, which is consistent with observations in prior works [24], [28]. This phenomenon may be attributed to the fact that middle and deep blocks incorporate more global context and thus capture more complex visual representations. Furthermore, it may provide some insight towards the construction of efficient ViTs. For example, we can maintain narrower channels in shallow blocks of ViTs.

7) Visualization of Consistencies: We conduct visualization analyses to show the positive effects of our proposed head-level consistency and attention-level consistency in CDCP. Specifically, we visualize the near-zero importance channels in each head and the ultimately pruned channels in the MHSA module, based on the DeiT-Tiny model with three heads. As mentioned in Section III-C, directly applying the conventional compactor pruning strategy for ViTs will cause imbalance ratios of pruned channels among heads and inconsistent pruned channels between query and key transformation matrices, leading to difficulties for parallel and error-free self-attention computation. Then, only the minimum ratio of near-zero importance channels among heads can be pruned and only the consistent near-zero importance channels between query and key transformation matrices can be removed. However, as shown in Fig. 7.(a), such a strategy will cause a substantial number of channels close to zero importance are retained in the compressed model, which impacts the performance adversely (Table XI). In contrast, our head-level and attention-level consistencies can maintain different heads

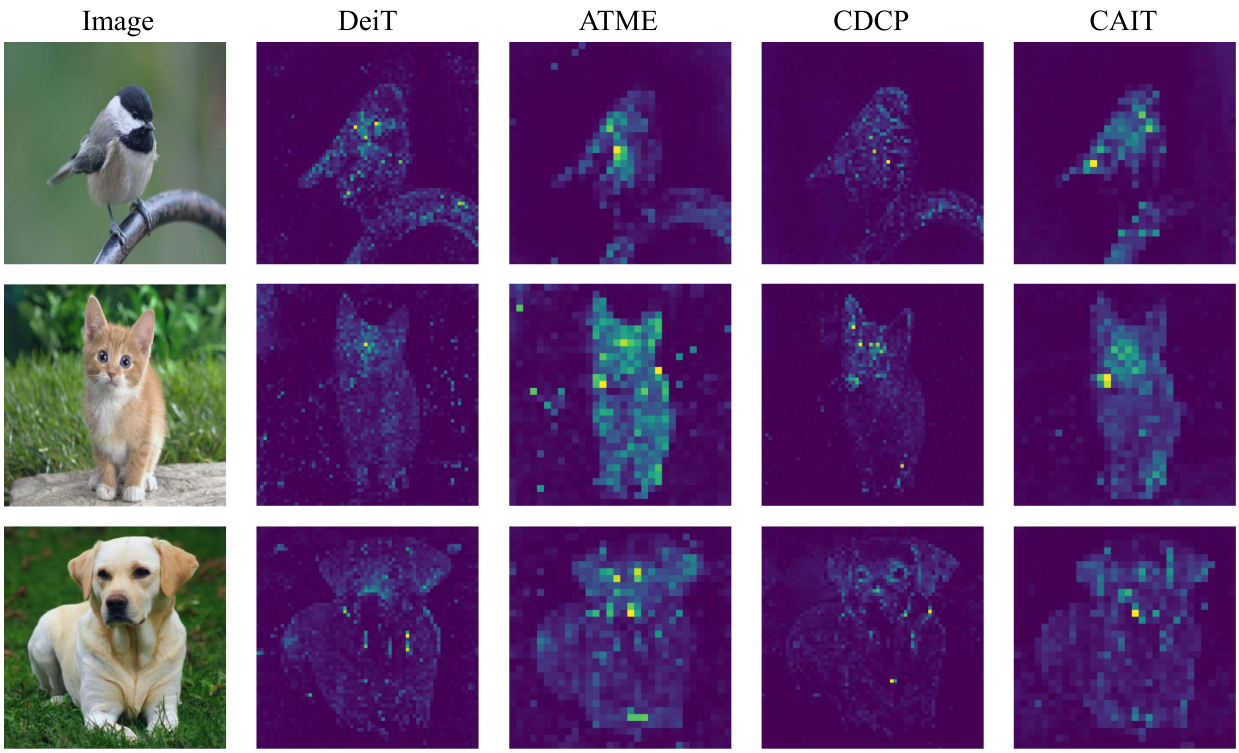

Fig. 8. Attention map comparison with token pruning by ATME, channel pruning by CDCP, and the joint pruning by CAIT.

of the same block in the same shape, and well align remained channels in the query and key, respectively. Therefore, as shown in Fig. 7.(b), our proposed consistencies can well address the limitations of vanilla compactor pruning on ViTs, ensuring error-free parallel self-attention computation and leading to superior performance (Table XI). Besides, as shown in Fig. 7, we can also observe that our proposed consistencies result in more pruned channels in MHSA modules and less pruned channels in FFN modules. This may be attributed to the fact that we encourage consistent shapes of different heads and aligned channels of the query and key in MHSA during pruning. It is also consistent with observations in previous works [27], [97], [98] that more redundant channels lie in MHSA modules. Additionally, results in Table 5 in the paper demonstrate the effectiveness of such a pruning strategy.

8) Visualization of Attention Maps: We conduct visualization analyses to inspect the impact caused by token pruning and channel pruning for attention maps based on DeiT-Small. Following [99], we visualize the attention map of the `[CLS]` token at the last layer. As shown in Fig. 8, the attention maps can be well preserved after token pruning by ATME, channel pruning by CDCP, and the joint pruning by CAIT. Besides, after token pruning, the important visual areas with highly semantic information can be strengthened due to the eliminated data level (*i.e.*, tokens) redundancy. Furthermore, thanks to the reduced model level (*i.e.*, parameters) redundancy by CDCP, the noise in the less informative regions can be well suppressed due to the less disturbance during computing attention. These favorable properties well help model to grasp critical visual information better, leading to improved efficiency.

V. Conclusion

In this paper, we proposes CAIT, a joint compression method with asymmetric token merging and consistent dynamic channel pruning for ViTs. The proposed asymmetric token merging strategy can effectively reduce the number of tokens while maintaining the spatial structure of images. The consistent dynamic channel pruning strategy can perform dynamic fine-grained compression optimization for all modules in ViTs. Extensive experiments on multiple ViTs over various vision tasks show that our method can outperform state-of-the-arts, achieving high performance, fast inference speed, and favorable transferability at the same time, well demonstrating its effectiveness and superiority.

Limitations. Although our CAIT shows superior performance and efficiency, it falls short of original models under certain scenarios and fails to achieve lossless compression. Besides, its transferability to 3D tasks and video tasks is also worth exploring. We leave these for future work.

References

[1] A. Dosovitskiy et al., "An image is worth 16 × 16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy
[2] K. Han et al., "A survey on vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023.
[3] T. Yao, Y. Li, Y. Pan, Y. Wang, X.-P. Zhang, and T. Mei, "Dual vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10870–10882, Sep. 2023.
[4] W. Yu et al., "Metaformer baselines for vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 2, pp. 896–912, Feb. 2024.

[5] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[6] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10 357.

[7] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose++: Vision transformer for generic body pose estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, pp. 1212–1230, Feb. 2024.

[8] Z. Guo, Z. Gu, B. Zheng, J. Dong, and H. Zheng, "Transformer for image harmonization and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12960–12977, Nov. 2023.

[9] Y.-H. Wu, Y. Liu, X. Zhan, and M.-M. Cheng, "P2T: Pyramid pooling transformer for scene understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12760–12771, Nov. 2023.

[10] Y. Song, Z. He, H. Qian, and X. Du, "Vision transformers for single image dehazing," *IEEE Trans. Image Process.*, vol. 32, pp. 1927–1941, 2023.

[11] J. Xiao, X. Fu, A. Liu, F. Wu, and Z.-J. Zha, "Image de-raining transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12978–12995, Nov. 2023.

[12] H. Chen et al., "Pre-trained image processing transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12299–12 310.

[13] J. Wei et al., "Emergent abilities of large language models," *Trans. Mach. Learn. Res.*, survey Certification, 2022. [Online]. Available: https://openreview.net/forum?id=yzkSU5zdwD

[14] A. Chowdhery et al., "Palm: Scaling language modeling with pathways," *J. Mach. Learn. Res.*, vol. 24, no. 1, Jan. 2023.

[15] O. OpenAI, "GPT-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[16] X. Chen et al., "Pali: A jointly-scaled multilingual language-image model," in *11th Int. Conf. Learn. Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=mWVoBz4W0u

[17] C. Riquelme et al., "Scaling vision with sparse mixture of experts," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 8583–8595.

[18] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12104–12113.

[19] M. Dehghani et al., "Scaling vision transformers to 22 billion parameters," in *Proc. 40th Int. Conf. Mach. Learn., ser.Proc. Mach. Learn. Res.*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., Jul. 2023, pp. 7480–7512. [Online]. Available: https://proceedings.mlr.press/v202/dehghani23a.html

[20] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4015–4026.

[21] T. Chen, Y. Cheng, Z. Gan, L. Yuan, L. Zhang, and Z. Wang, "Chasing sparsity in vision transformers: An end-to-end exploration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 19974–19 988.

[22] Z. Kong et al., "Spvit: Enabling faster vision transformers via latency-aware soft token pruning," in *Proc. Comput. Vis., 17th Eur. Conf.*, 2022, pp. 620–640.

[23] S. Yu et al., "Unified visual transformer compression," in *Proc. Int. Conf. Learn. Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=9jsZiUgkCZP

[24] C. Zheng et al., "Savit: Structure-aware vision transformer pruning via collaborative optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 9010–9023.

[25] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh, "Dynamicvit: Efficient vision transformers with dynamic token sparsification," *Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 13937–13949w.

[26] M. Zhu, Y. Tang, and K. Han, "Vision transformer pruning," 2021, *arXiv:2104.08500*.

[27] H. Yang, H. Yin, P. Molchanov, H. Li, and J. Kautz, "Nvit: Vision transformer compression and parameter redistribution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 18 547–18 557.

[28] Z. Wang, H. Luo, P. Wang, F. Ding, F. Wang, and H. Li, "VTC-LFC: Vision transformer compression with low-frequency components," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 13974–13 988.

[29] Y. Tang et al., "Patch slimming for efficient vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12165–12174.

[30] Y. Liang, C. Ge, Z. Tong, Y. Song, J. Wang, and P. Xie, "EVit: Expediting vision transformers via token reorganizations," in *Proc. Int. Conf. Learn. Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=BjyvwnXXVn_

[31] Z. Hou and S.-Y. Kung, "Multi-dimensional model compression of vision transformer," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2022, pp. 1–6.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[33] B. Pan, R. Panda, Y. Jiang, Z. Wang, R. Feris, and A. Oliva, "IA-RED^2: Interpretability-aware redundancy reduction for vision transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 24898–24911.

[34] D. Bolya, C.-Y. Fu, X. Dai, P. Zhang, C. Feichtenhofer, and J. Hoffman, "Token merging: Your vit but faster," in *Proc. 11th Int. Conf. Learn. Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=JroZRaRw7Eu

[35] S. Wei, T. Ye, S. Zhang, Y. Tang, and J. Liang, "Joint token pruning and squeezing towards more aggressive compression of vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 2092–2101.

[36] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 418–434.

[37] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1290–1299.

[38] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[39] Z. Cai and N. Vasconcelos, "Cascade r-CNN: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1483–1498, May 2021.

[40] G. Cheng, A. Matsune, Q. Li, L. Zhu, H. Zang, and S. Zhan, "Encoder-decoder residual network for real super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 2169–2178.

[41] Y. Xu et al., "Evo-vit: Slow-fast token evolution for dynamic vision transformer," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 3, 2022, pp. 2964–2972.

[42] S. Chang et al., "Making vision transformers efficient from a token sparsification view," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 6195–6205.

[43] H. He et al., "Pruning self-attentions into convolutional layers in single path," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 3910–3922, 2024.

[44] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000–6010.

[45] J. Guo et al., "CMT: Convolutional neural networks meet vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12175–12 185.

[46] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: BERT pre-training of image transformers," in *Proc. Int. Conf. Learn. Representations*, 2022.

[47] L. Yuan et al., "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 558–567.

[48] C.-F. R. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 357–366.

[49] A. Ali et al., "XCiT: Cross-covariance image transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 20014–20027.

[50] B. Graham et al., "Levit: A vision transformer in convnet's clothing for faster inference," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12259–12269.

[51] P. Wang et al., "KVT: *k*-NN attention for boosting vision transformers," in *Proc. Comput. Vis., 17th Euro. Conf.*, 2022, pp. 285–302.

[52] X. Chen, C.-J. Hsieh, and B. Gong, "When vision transformers outperform resnets without pre-training or strong data augmentations," in *Proc. Int. Conf. Learn. Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=LtKcMgGOeLt

[53] A. Amini, A. S. Periyasamy, and S. Behnke, "T6D-direct: Transformers for multi-object 6D pose direct regression," in *Proc. Pattern Recognit., 43rd DAGM German Conf.*, Bonn, Germany, 2022, pp. 530–544.

[54] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=gZ9hCDWe6ke

[55] Z. Dai, B. Cai, Y. Lin, and J. Chen, "Up-detr: Unsupervised pre-training for object detection with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1601–1610.

[56] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2906–2917.

[57] S. He, H. Luo, P. Wang, F. Wang, H. Li, and W. Jiang, "Transreid: Transformer-based object re-identification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15013–150 22.
[58] A. El-Nouby, N. Neverova, I. Laptev, and H. Jégou, "Training vision transformers for image retrieval," 2021, *arXiv:2102.05644*.
[59] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 17864–17875.
[60] Y. Wang et al., "End-to-end video instance segmentation with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8741–8750.
[61] L. Ding et al., "Looking outside the window: Wide-context transformer for the semantic segmentation of high-resolution remote sensing images1," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022.
[62] S. Zheng et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6881–6890.
[63] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning texture transformer network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5791–5800.
[64] X. Lai et al., "Stratified transformer for 3D point cloud segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8500–8509.
[65] M. S. Ryoo, A. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova, "Tokenlearner: What can 8 learned tokens do for images and videos?," *Adv. Neural Inf. Process. Syst.*, 2021.
[66] J. Zhang et al., "Minivit: Compressing vision transformers with weight multiplexing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12145–12 154.
[67] Z. Zong et al., "Self-slimmed vision transformer," in *Proc. Comput. Vis., 17th Euro. Conf.*, 2022, pp. 432–448.
[68] Y. He, G. Kang, X. Dong, Y. Fu, and Y. Yang, "Soft filter pruning for accelerating deep convolutional neural networks," in *Proc. 27th Int. Joint Conf. Arti. Intell.*, 2018, pp. 2234–2240.
[69] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4340–4349.
[70] X. Ding et al., "Resrep: Lossless cnn pruning via decoupling remembering and forgetting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4510–4520.
[71] S. Lin, R. Ji, Y. Li, Y. Wu, F. Huang, and B. Zhang, "Accelerating convolutional networks via global & dynamic filter pruning," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 2425–2432. [Online]. Available: https://doi.org/10.24963/ijcai.2018/336
[72] Z. Hou et al., "Chex: Channel exploration for CNN model compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12287–12 298.
[73] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky, "Sparse convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 806–814.
[74] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA: Curran Associates Inc., 2016, pp. 2082–2090.
[75] "fvcore library," 2019. [Online]. Available: https://github.com/facebookresearch/fvcore/
[76] Z. Song, Y. Xu, Z. He, L. Jiang, N. Jing, and X. Liang, "Cp-vit: Cascade vision transformer pruning via progressive sparsity prediction," 2022, *arXiv:2203.04570*.
[77] J. Ansel et al., "PyTorch 2: Faster machine learning through dynamic Python bytecode transformation and graph compilation," in *Proc. 29th ACM Int. Conf. Architectural Support Program. Lang. Operating Syst.*, vol. 2, Apr. 2024. [Online]. Available: https://docs.pytorch.org/assets/pytorch2-2.pdf
[78] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6399–6408.
[79] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20 k dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 633–641.
[80] Z.-H. Jiang et al., "All tokens matter: Token labeling for training better vision transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 18590–18602.
[81] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7
[82] M. Contributors, "Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark," 2020. [Online]. Available: https://github.com/open-mmlab/mmsegmentation
[83] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16000–16009.
[84] K. Tian, Y. Jiang, Q. Diao, C. Lin, L. Wang, and Z. Yuan, "Designing bert for convolutional networks: Sparse and hierarchical masked modeling," in *Proc. 11th Int. Conf. Learn. Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=NRxydtWup1S
[85] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.
[86] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Comput. Vis., 13th Euro. Conf.*, 2014, pp. 740–755.
[87] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 568–578.
[88] M. M. Rahman and R. Marculescu, "Medical image segmentation via cascaded attention decoding," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 6222–6231.
[89] B. Landman, Z. Xu, J. Igelsias, M. Styner, T. Langerak, and A. Klein, "Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge," in *Proc. Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge*, Munich, Germany, 2015, vol. 5, p. 12.
[90] O. Bernard et al., "Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: Is the problem solved?," *IEEE Trans. Med. Imag.*, vol. 37, no. 11, pp. 2514–2525, Nov. 2018.
[91] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodrıguez, and F. Vilariño, "WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians," *Computerized Med. Imag. Graph.*, vol. 43, pp. 99–111, 2015.
[92] D. Jha et al., "Kvasir-SEG: A segmented polyp dataset," in *Proc. MultiMedia Model., 26th Int. Conf.*, Daejeon, South Korea, Jan. 5–8, 2020, pp. 451–462.
[93] D. Wang, J. Zhang, B. Du, G.-S. Xia, and D. Tao, "An empirical study of remote sensing pretraining," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5608020.
[94] "2D semantic labeling contest - Potsdam," Int. Soc. Photogrammetry Remote Sens, 2021. [Online]. Available: https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx
[95] S. Waqas Zamir et al., "iSAID: A large-scale dataset for instance segmentation in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 28–37.
[96] M. Chen et al., "Diffrate: Differentiable compression rate for efficient vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 17164–17174.
[97] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan, "Efficientvit: Memory efficient vision transformer with cascaded group attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023.
[98] W. Yu et al., "Metaformer is actually what you need for vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10819–10 829.
[99] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9650–9660.

Ao Wang (Graduate Student Member, IEEE) received the BE degree in 2022 from the School of Software, Tsinghua University, China, where he is currently working toward the PhD degree with the School of Software. His research interests include computer vision and machine learning.

Hui Chen is currently an assistant researcher with the Beijing National Research Center for Information Science and Technology, Tsinghua University. He has authored or coauthored more than 15 peer-reviewed top conference and journal papers, including CVPR, ICCV, and ICLR. His research focuses on efficient and effective multi-modal perception and learning. He was a PC member of several top-tier conferences.

Zijia Lin received the BSc degree from the School of Software, and the PhD degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2011 and 2016, respectively. His research interests include multimedia information retrieval and machine learning.

Sicheng Zhao (Senior Member, IEEE) received the PhD degree from the Harbin Institute of Technology, Harbin, China, in 2016. He was a visiting scholar with the National University of Singapore, Singapore, from 2013 to 2014, research fellow with Tsinghua University, Beijing, China, from 2016 to 2017, postdoctoral research fellow with the University of California at Berkeley, Berkeley, CA, USA, from 2017 to 2020, and postdoctoral research scientist with Columbia University, New York, NY, USA, from 2020 to 2022. He is currently a research associate professor with Tsinghua University. His research interests include affective computing, multimedia, and computer vision.

Jungong Han (Senior Member, IEEE) is currently a tenured professor with the Department of Automation, Tsinghua University. He also holds an honorary professorship with the University of Warwick, U.K. He has authored two edited volumes, and more than 200 papers, including 90 in prestigious IEEE/ACM Transactions, and more than 60 in CORE A* conferences. His research interests include computer vision and multi-modal learning. He is a fellow of IAPR and AAIA.

Guiguang Ding (Senior Member, IEEE) is currently a tenured professor with the School of Software, Tsinghua University. He has authored or coauthored more than 30 papers in top-tier journals including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Signal Processing Magazine*, and *IEEE Transactions on Image Processing*. His reasearch interests include model architecture design and compression, visual semantic recognition and description, transfer learning, and few-shot learning. He has presented more than 70 papers at top-tier international conferences, such as CVPR, NeurIPS, and ICML.