

# CF-ViT: A General Coarse-to-Fine Method for Vision Transformer

Mengzhao Chen<sup>1</sup>, Mingbao Lin<sup>3</sup>, Ke Li<sup>3</sup>, Yunhang Shen<sup>3</sup>,  
Yongjian Wu<sup>3</sup>, Fei Chao<sup>1</sup>, Rongrong Ji<sup>1,2\*</sup>

<sup>1</sup>MAC Lab, Department of Artificial Intelligence, Xiamen University

<sup>2</sup>Institute of Artificial Intelligence, Xiamen University

<sup>3</sup>Tencent YouTu Lab

cmzxm@stu.xmu.edu.cn, linmb001@outlook.com, tristanli.sh@gmail.com,  
shenyunhang01@gmail.com, littlekenwu@tencent.com, {fchao, rrji}@xmu.edu.cn

## Abstract

Vision Transformers (ViT) have made many breakthroughs in computer vision tasks. However, considerable redundancy arises in the spatial dimension of an input image, leading to massive computational costs. Therefore, We propose a coarse-to-fine vision transformer (CF-ViT) to relieve computational burden while retaining performance in this paper. Our proposed CF-ViT is motivated by two important observations in modern ViT models: (1) The coarse-grained patch splitting in modern ViT models: (1) The coarse-grained patch splitting can locate informative regions of an input image. (2) Most images can be well recognized by a ViT model in a small-length token sequence. Therefore, our CF-ViT implements network inference in a two-stage manner. At coarse inference stage, an input image is split into a small-length patch sequence for a computationally economical classification. If not well recognized, the informative patches are identified and further re-split in a fine-grained granularity. Extensive experiments demonstrate the efficacy of our CF-ViT. For example, without any compromise on performance, CF-ViT reduces 53% FLOPs of LV-ViT, and also achieves  $2.01\times$  throughput. Code of this project is at <https://github.com/ChenMnZ/CF-ViT>.

## Introduction

Tremendous successes of traditional transformer (Vaswani et al. 2017) in natural language processing (NLP) have inspired the researchers to go further on computer vision (Han et al. 2022a). Consequently, vision transformers (ViT) (Dosovitskiy et al. 2020) receive ever-increasing attentions in many vision tasks such as image classification (Dosovitskiy et al. 2020; Jiang et al. 2021), object detection (Liu et al. 2021; Wang et al. 2021a), semantic segmentation (Zheng et al. 2021; Xie et al. 2021), *etc.*

By splitting a 2D image into a patch sequence and using a linear projection to embed these patches into 1D tokens as inputs, ViT merits in its property of modeling long-range dependencies among tokens. Generally speaking, the performance of a ViT model is closely correlated with the token number (Dosovitskiy et al. 2020; Wang et al. 2021c), to which, however, the computational cost of ViT is also quadratically related. Fortunately, images often have more spatial redundancy than languages (Wang, Stuijk, and

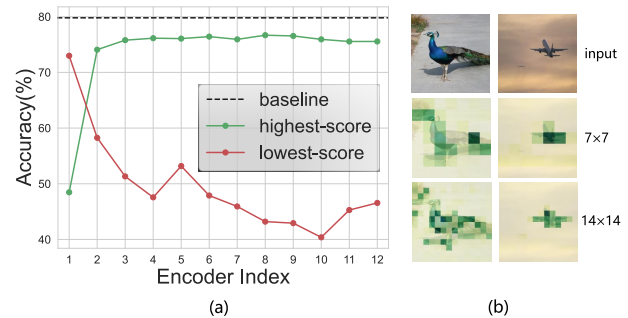


Figure 1: (a) Performance comparison (DeiT-S (Touvron et al. 2021a)) between 100 highest-score patches and 100 lowest-score patches. The score is measured by the class attention. (b) Visualization of class attention in the last encoder of DeiT-S (Best viewed with zooming in).

De Haan 2014), such as regions with task-unrelated objects. Thus, many works (Wang et al. 2020; Yang et al. 2020; Wang et al. 2022a, 2021b, 2022b; Han et al. 2022b, 2021c) try to adaptively reduce the input resolution of convolution neural networks. Also, great efforts have been made to excavate redundant tokens for ViTs. For example, PS-ViT (Tang et al. 2022b) enhances transformer efficiency by a top-down token pruning paradigm. Both DynamicViT (Rao et al. 2021) and IA-RED<sup>2</sup> (Pan et al. 2021a) devise a lightweight prediction module to estimate the importance score of each token, and discard low-score tokens. Following DynamicViT (Rao et al. 2021), EViT (Liang et al. 2022) regards class attention as a metric of token importance, which avoids the introduction of extra parameters. Unlike discarding tokens directly, DGE (Song et al. 2021) introduces sparse queries to reduce the output token number. Evo-ViT (Xu et al. 2022) maintains the spatial structure while consuming less computational cost to update uninformative tokens. DVT (Wang et al. 2021c) cascades multiple ViTs with increasing tokens, then leverages an early-exiting policy to decide each image’s token number.

This paper observes two insightful phenomena. First, similar to the fine-grained patch splitting like  $14\times 14$ , the coarse-grained patch splitting such as  $7\times 7$  can also locate the informative regions. To verify this, we first show that

\*Corresponding Author

No. of token	$14 \times 14$	$7 \times 7$
Accuracy	79.8%	73.2%
FLOPs	4.60G	1.10G

Table 1: Accuracy and FLOPs of DeiT-S (Touvron et al. 2021a) on ImageNet with different No. of patches as inputs.

class attention (see Eq. (2)) (Liang et al. 2022) has a remarkable ability to identify informative patches. We conduct a toy experiment on the validation set of ImageNet (Deng et al. 2009) with a pre-trained DeiT-S model (Touvron et al. 2021a). Following DeiT-S, we split each image into  $14 \times 14$  patches, then compute the class attention score of each patch within each encoder. Then, the 100 highest-score patches and 100 lowest-score patches are respectively fed to DeiT-S to obtain their accuracy in Fig. 1(a). In the figure, highest-score patches outperform lowest-score ones by a large margin, such a result demonstrates that class attention can well reflect more informative patches. Then, in Fig. 1(b), we visualize the class attention in the last encoder of DeiT-S. As can be seen,  $7 \times 7$  splitting and  $14 \times 14$  splitting generally obtain similar attentive regions.

The second phenomenon is that most images can be well recognized by a ViT model in a small-length patch sequence. We train DeiT-S (Touvron et al. 2021a) with varying lengths of input sequence, and report top-1 accuracy and FLOPs in Table 1, in which, with  $4.2 \times$  higher computational cost, splitting a 2D image into fine-grained  $14 \times 14$  patches only obtains 6.6% accuracy benefits than coarse-grained  $7 \times 7$  splitting. A similar observation has been discussed in DVT (Wang et al. 2021c). This indicates that most regions in 73.2% images are “easy” so that coarse-grained  $7 \times 7$  patch splitting can well implement the classification, and a small portion of images are filled with “hard” regions requiring a fine-grained splitting of  $14 \times 14$  with heavier computation. Thus, we split the “easy” samples with coarse-grained patch splitting for cheaper computation. In addition, for “hard” samples, we can differentiate “easy” regions and “hard” regions, and split them with different patch sizes in order to pursue an efficient inference while retaining good performance. Note that, these two phenomena can be found in other ViT models as well.

Inspired by the above observations, we propose a novel coarse-to-fine vision transformer in this paper, termed CF-ViT, which aims to produce correct predictions with input-adaptive computational cost. As shown in Fig. 2, the inference of CF-ViT is divided into a coarse inference stage and a fine inference stage. The coarse stage receives coarse-grained patches as network inputs, which merits in low computational cost since the coarse-grained splitting results in much fewer patches (tokens). If this stage owns a high confidence score, the network inference terminates; otherwise, the input image is split again into fine-grained patches and fed to the fine stage. In contrast to a naive re-splitting of all coarse patches, we design a mechanism of informative region identification to further split the informative patches in a fine-grained state, and retain these patches with less information in a coarse-grained state. Therefore, we avoid heavy

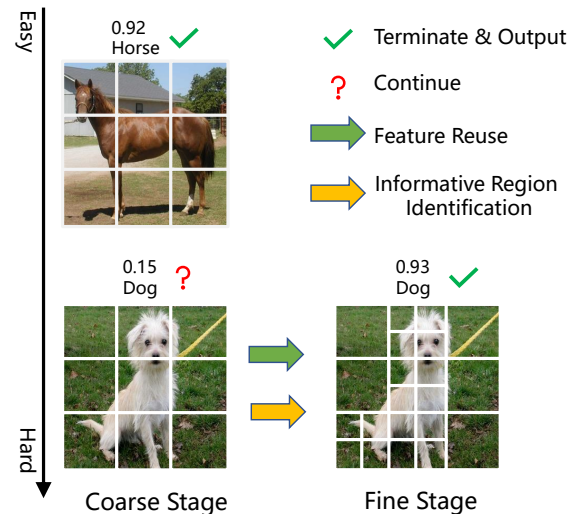


Figure 2: Example of CF-ViT.

computational burdens on the redundant image patches. Also, we introduce a mechanism of feature reuse to inject the integral information of the coarse patch to the split fine-grained patches, which further enhance the model performance. We then evaluate our proposed CF-ViT built upon DeiT (Touvron et al. 2021a) and LV-ViT (Jiang et al. 2021) on ImageNet (Deng et al. 2009). Extensive experiments results show that our CF-ViT can well boost the inference efficiency. For example, without any performance compromise, CF-ViT reduces 53% FLOPs of LV-ViT and also leads to  $2.01 \times$  practical throughput on an A100 GPU. In addition, extensive ablation studies also demonstrate the efficacy of each design in our CF-ViT including the informative region identification and the feature reuse. Visualization of inference results shows that CF-ViT enables adaptive inference according to the “difficulty” of input images and can accurately locate informative regions for re-splitting in fine inference stage.

## Related Work

### Vision Transformer

Motivated by successes of transformer (Vaswani et al. 2017) in NLP, researchers develop vision transformer (ViT) (Dosovitskiy et al. 2020) for image recognition. However, the lack of inductive bias (Dosovitskiy et al. 2020) requires ViT model to be pre-trained on a very large-scale data corpus such as JFT-300M (Sun et al. 2017) to pursue a desired performance. This demand barricades the development of ViT model since the large-scale dataset requires a high-capacity workstation. To handle this, many studies (Touvron et al. 2021a; Jiang et al. 2021) develop specialized training strategies for ViT models. For instance, DeiT (Touvron et al. 2021a) introduces an extra token for knowledge distillation. LV-ViT (Jiang et al. 2021) leverages all tokens to compute the training loss, and the location-specific supervision label of each patch token is generated by a machine anno-

tator. In addition, another group focuses on improving the architecture of ViT (Yuan et al. 2021a; Li et al. 2021; Chu et al. 2021b,a; Yuan et al. 2021b; Heo et al. 2021; Touvron et al. 2021b; Liu et al. 2021; Li et al. 2022). For example, CPVT (Chu et al. 2021b) uses a convolution layer to replace the learnable positional embedding. CaiT (Touvron et al. 2021b) builds deeper transformers and develops specialized optimization strategies for training. TNT (Han et al. 2021a) leverages an inner block to model the pixel-wise interactions within each patch, which merits in preserving more rich local features. Our study in this paper introduces a general framework, which aims to improve the inference efficiency of various ViT backbones in an input-adaptive manner.

### ViT Compression

Except to develop high-performing ViT models, the expensive computation of a ViT model also arouses wide attention and methods are explored to facilitate ViT deployment. Based on whether the inference is input-dependent, we empirically categorize existing studies on compressing ViTs into two groups below.

**Static ViT Compression.** This group mainly focuses on reducing the network complexity through manually designed modules with a fixed computational graph regardless of the input images. Inspired by the great success of hierarchical convolutional neural networks in dense prediction tasks such as segmentation and detection, recent advances (Heo et al. 2021; Liu et al. 2021; Pan et al. 2021b; Yuan et al. 2021b; Wang et al. 2021a) introduce hierarchical transformers. Also, many others (Liu et al. 2021; Huang et al. 2021; Fang et al. 2021; Yu et al. 2021) consider local self-attention to reduce the complexity of traditional global self-attention. Compared to these static methods using a fixed computational graph, our CF-ViT, improves the inference efficiency by adaptively selecting an appropriate computational path for each image.

**Dynamic ViT Compression.** In contrast to static ViT compression, dynamic ViT adapts the computational graph according to its input images (Han et al. 2021b). Some works (Rao et al. 2021; Pan et al. 2021a; Liang et al. 2022; Lin et al. 2022) attempt to dynamically prune these tokens considered unimportant during inference. On the contrary, Evo-ViT (Xu et al. 2022) chooses to preserve the unimportant tokens, which however, are assigned with a lower computational budget for updating. Unlike these pruning-based methods, our CF-ViT maintains the integrity of image information and reinforces the inference efficiency by enlarging the size of patches in uninformative regions, which leads to fewer tokens. The rationale behind this is that uninformative regions like background contribute less to the recognition thus a fine-grained patch splitting is unnecessary. Note that, the recent DVT (Wang et al. 2021c) endows a proper token number for each input image by cascading three transformers. Though meriting in inference acceleration, it inevitably increases the storage overhead by  $3\times$ . Different from DVT, our CF-ViT trains only one transformer that can accept different sizes of input tokens, and only conducts fine-grained token splitting on informative regions rather than an entire image. QuadTree (Tang et al. 2022a) also builds token pyra-

mids in a coarse-to-fine manner. However, QuadTree performs coarse-to-fine splitting for all images while our CF-ViT only performs fine splitting on “hard” images to further reduce computation cost.

### Preliminaries

Vision Transformer (ViT) (Dosovitskiy et al. 2020) splits a 2D image into flattened 2D patches and uses a linear projection to map patches into tokens, *a.k.a.* patch embeddings. Besides, an extra [class] token, which represents the global image information, is appended as well. Moreover, all tokens are added with a learnable positional embedding. Thus, the input token sequence of a ViT model is:

$$\mathbf{X}_0 = [\mathbf{x}_0^0; \mathbf{x}_0^1; \dots; \mathbf{x}_0^N] + \mathbf{E}_{pos}, \quad (1)$$

where  $\mathbf{x}_0^i \in \mathbb{R}^D$  is a  $D$ -dimensional token of the  $i$ -th patch if  $i > 0$ , and [class] token if  $i = 0$ . The  $\mathbf{E}_{pos}$  and  $N$  are the position embedding and patch number.

A ViT model  $\mathcal{V}$  contains  $K$  sequentially stacked encoders, each of which consists of a self-attention (SA) module<sup>1</sup> and a feed-forward network (FFN). In SA of the  $k$ -th encoder, the token sequence  $\mathbf{X}_{k-1}$  is projected into a query matrix  $\mathbf{Q}_k \in \mathbb{R}^{(N+1) \times D}$ , a key matrix  $\mathbf{K}_k \in \mathbb{R}^{(N+1) \times D}$ , and a value matrix  $\mathbf{V}_k \in \mathbb{R}^{(N+1) \times D}$ . Then, the self-attention matrix  $\mathbf{A}_k \in \mathbb{R}^{(N+1) \times (N+1)}$  is computed as:

$$\mathbf{A}_k = \text{Softmax}\left(\frac{\mathbf{Q}_k \mathbf{K}_k^T}{\sqrt{D}}\right) = [\mathbf{a}_k^0; \mathbf{a}_k^1; \dots; \mathbf{a}_k^N]. \quad (2)$$

The  $\mathbf{a}_k^0 \in \mathbb{R}^{(N+1)}$  is known as class attention, reflecting the interactions between [class] token and other patch tokens. With  $\mathbf{A}_k$ , the outputs of SA, *i.e.*,  $\mathbf{A}_k \mathbf{V}_k$ , are sent to FFN consisting of two fully-connected layers to derive the updated tokens  $\mathbf{X}_k = [\mathbf{x}_k^0; \mathbf{x}_k^1; \dots; \mathbf{x}_k^N]$ . The [class] token  $\mathbf{x}_k^0$  is derived as:

$$\mathbf{x}_k^0 = \text{FFN}(\mathbf{a}_k^0 \mathbf{V}_k). \quad (3)$$

After a series of SA-FFN transformations, the [class] token  $\mathbf{x}_K^0$  from the  $K$ -th encoder is fed to the classifier to predict the category of the input.

**ViT Complexity.** Given that an image is split into  $N$  patches, the computational complexity of SA and FFN are (Xu et al. 2022):

$$\begin{aligned} O(SA) &= 3ND^2 + 2N^2D, \\ O(FFN) &= 8ND^2. \end{aligned} \quad (4)$$

We can see that the complexities of SA and FNN are respectively quadratic and linear to  $N$ . Thus, ViT complexity can be well reduced by decreasing the input patch (token) number (Rao et al. 2021; Pan et al. 2021a; Liang et al. 2022; Wang et al. 2021c), which is also the focus of this paper.

<sup>1</sup>SA has been replaced by multi-head self-attention (MHSA) in most ViTs. For brevity, we simply discuss SA herein. Also, we ignore the shortcut operation in Eq. (3).

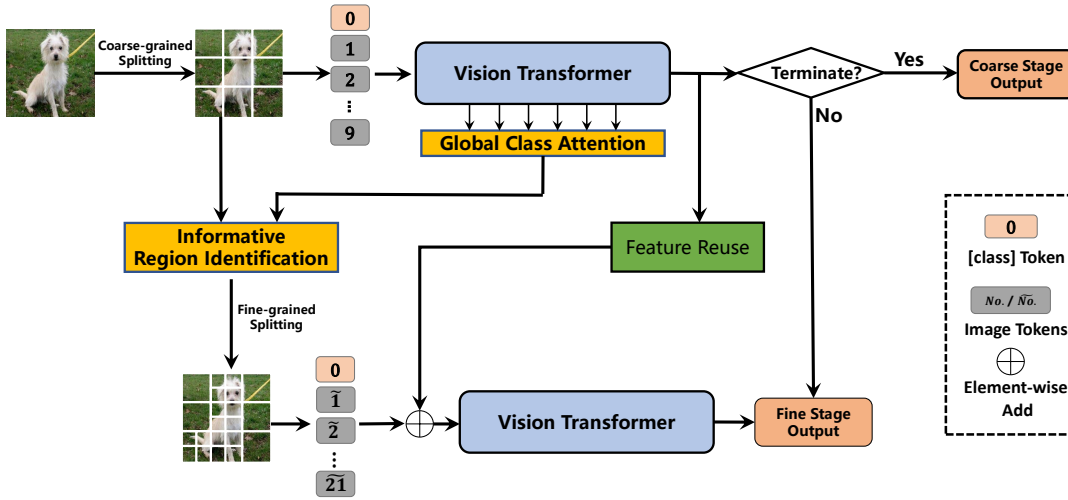


Figure 3: Framework of our CF-ViT. Note that both coarse stage and fine stage share the same network parameters.

### Coarse-to-Fine Vision Transformer

This section formally introduces our CF-ViT that decreases the computational cost by reducing the input sequence length. Our motive lies in two observations in Sec. : (1) The coarse-grained patch splitting can well locate the informative objects as well. (2) Most images can be well recognized by a ViT model in a small sequence length. These inspire us to implement a ViT in a two-stage manner. As shown in Fig. 3, the coarse inference stage implements image recognition with a small length of token sequence. If not well recognized, the informative regions will be further split for a fine-grained recognition. Details are given below.

#### Coarse Inference Stage

CF-ViT first performs a coarse splitting to recognize images filled with “easy” regions. Also, it locates informative regions for an efficient inference when meeting “hard” samples. At coarse stage, the input of our CF-ViT model  $\mathcal{V}$  is:

$$\mathbf{X}_0^c = [\mathbf{x}_0^0; \mathbf{x}_0^1; \dots; \mathbf{x}_0^{N_c}] + \mathbf{E}_{pos}^c, \quad (5)$$

where  $N_c$  is the number of coarse patches. Supposing  $\mathcal{V}$  contains  $K$  encoders, after SA-FFN transformations in Sec. , the output token sequence of  $\mathcal{V}$  is:

$$\mathcal{V}(\mathbf{X}_0^c) = [\mathbf{x}_K^0; \mathbf{x}_K^1; \dots; \mathbf{x}_K^{N_c}]. \quad (6)$$

Finally, the [class] token  $\mathbf{x}_K^0$  is fed to a classifier  $\mathcal{F}$  to obtain the coarse-stage category prediction distribution  $\mathbf{p}^c$ :

$$\mathbf{p}^c = \mathcal{F}(\mathbf{x}_K^0) = [p_1^c, p_2^c, \dots, p_n^c], \quad (7)$$

where  $n$  denotes the category number. So far, we can obtain the predicted category of the input as:

$$j = \arg \max_i p_i^c. \quad (8)$$

We expect a large  $p_j^c$  since it serves as a prediction confidence score at coarse inference stage where the computational cost is very cheap due to a small value of patch

number  $N_c$ . We introduce a threshold  $\eta$  to realize a trade-off between performance and computation. In our implementation, if  $p_j^c \geq \eta$ , the inference will terminate and we attribute the input to category  $j$ . Otherwise, the input images might contain “hard” regions undistinguished to the ViT model. Thus, a more fine-grained patch splitting is urgent.

**Informative Region Identification.** The most naive solution is to further split all the coarse patches  $[\mathbf{x}_0^1; \mathbf{x}_0^2; \dots; \mathbf{x}_0^{N_c}]$ . However, the drastically increasing tokens inevitably cause severe computational costs. Instead, for an economical budget, we propose to identify and then re-split these informative regions that are the most beneficial to the performance increase. Thus, the key now lies in how to identify the informative patches.

Recall that, the class attention  $\mathbf{a}_k^0 \in \mathbf{A}_k$  in Eq. (2) reflects the interactions between [class] token and other image patch tokens in the  $k$ -th encoder. Besides, the [class] token  $\mathbf{x}_k^0 = FFN(\mathbf{a}_k^0 \mathbf{V}_k)$ , which indicates that each item  $(\mathbf{a}_k^0)_i$  models the weighted coefficient of the  $i$ -th token  $\mathbf{x}_k^i$  to the performance. Therefore, it is natural to use the class attention  $\mathbf{a}_k^0$  as a score to indicate if a token is informative. From Fig. 1(a), we can see that, the performance of preserving 100 highest-score patches usually outperforms that of 100 lowest-score patches. This demonstrates that class attention can be a reliable measure. Nevertheless, we can also observe the instability of class attention in the bottom layers, such as the first encoder. To overcome it, we propose global class attention that combines class attention across different encoders using exponential moving average (EMA) to better identify informative patches:

$$\bar{\mathbf{a}}_k = \beta \cdot \bar{\mathbf{a}}_{k-1} + (1 - \beta) \cdot \mathbf{a}_k^0, \quad (9)$$

where  $\beta = 0.99$ . The global class attention begins from the 4-th encoder and we select patches with high-score global class attention in the last encoder  $\bar{\mathbf{a}}_K$ .

Earlier studies (Liang et al. 2022; Xu et al. 2022) also adopt class attention to indicate token importance. This paper differs in two folds: (1) A comprehensive demonstration

on high-score patches are given in Fig. 1(a). (2) We consider the global class attention instead of class attention in a particular layer (Liang et al. 2022), efficacy of which is given in Table. 4.

### Fine Inference Stage

With the global class attention  $\bar{\mathbf{a}}_K$  on hand, we continue to perform a fine-grained splitting on informative patches when the prediction in coarse stage  $p_j^c < \eta$ , which indicates an indistinguishable input image.

To that effect, we pick up these coarse patches, attention scores of which are within the top- $\alpha N_c$  largest among the  $N_c$  coarse patches, where  $\alpha \in [0, 1]$  represents the rate of informative patches. Then, as shown in Fig. 3, each informative patch is further split into  $2 \times 2$  patches for better representation in a finer granularity. Consequently, the patch number after fine-grained splitting is:

$$N_f = 4\lceil N_c \alpha \rceil + \lfloor N_c(1 - \alpha) \rfloor, \quad (10)$$

where  $\lceil \cdot \rceil$  and  $\lfloor \cdot \rfloor$  respectively round up and round down their inputs. It is intuitive that  $\alpha$  provides a trade-off between accuracy and efficiency. The  $\alpha = 0$  indicates no fine inference and results in the fewest patches. Though computationally economical, performance drops if the test set is full of “hard” images. In contrast,  $\alpha = 1$  leads the fine inference stage of our CF-ViT degenerates to traditional ViT models (Jiang et al. 2021; Touvron et al. 2021a). In this case, the computational cost is extraordinarily expensive. For a trade-off, we set  $\alpha$  to 0.5 in our implementation.

**Feature Reuse.** After re-splitting of each input image, the input token sequence at fine inference stage of our CF-ViT becomes:

$$\tilde{\mathbf{X}}_0^f = [\mathbf{x}_0^0; \tilde{\mathbf{x}}_0^1; \dots; \tilde{\mathbf{x}}_0^{N_f}] + \mathbf{E}_{pos}^f. \quad (11)$$

Suppose that an informative patch  $\mathbf{x}_0^i$  in Eq. (5) is further split into  $2 \times 2$  patches in Eq. (11), which offer a finer granularity. Nevertheless, they also cut off the integrity of the local patch  $\mathbf{x}_0^i$ . To solve this, we also devise a feature reuse module to inject the information of  $\mathbf{x}_0^i$  into the four fine-grained patches.

Fig. 4 illustrates our feature reuse. It takes the output token sequence from coarse stage as input. Similar to FNN, the input will be processed by an MLP layer first to allow a flexible transformation. Then, these transformed tokens are reshaped and each of them is copied  $4 \times$ . Further, these tokens corresponding to fine-grained splitting patches are picked up as the output of feature reuse module, denoted as  $\mathbf{X}_r = FR([\mathbf{x}_K^1; \mathbf{x}_K^2; \dots; \mathbf{x}_K^{N_c}])$ . We do not consider reusing [class] token and uninformative tokens by zeroing out them, since we empirically find them unbeneficial to the performance as verified in Table. 5. Finally, we shortcut  $\mathbf{X}_r$  to the re-split token sequence  $\tilde{\mathbf{X}}_0^f$  in Eq. (11) as the final input of our CF-ViT model  $\mathcal{V}$  at fine inference stage. Consequently, the output of  $\mathcal{V}$  is:

$$\mathcal{V}(\tilde{\mathbf{X}}_0^f + \mathbf{X}_r) = [\tilde{\mathbf{x}}_K^0; \tilde{\mathbf{x}}_K^1; \dots; \tilde{\mathbf{x}}_K^{N_f}]. \quad (12)$$

Finally, the [class] token  $\tilde{\mathbf{x}}_K^0$  is fed to the same classifier  $\mathcal{F}$  to obtain the fine-stage category prediction distribution  $\mathbf{p}^f$ :

$$\mathbf{p}^f = \mathcal{F}(\tilde{\mathbf{x}}_K^0) = [p_1^f, p_2^f, \dots, p_n^f]. \quad (13)$$

Model	$\eta$	Top-1 Acc. (%)	FLOPs (G)	Throughput (img./s)
DeiT-S	-	79.8	4.6	2601
CF-ViT	0.5	79.8(+0.0)	1.8(↓ 61%)	4903(↑1.88×)
CF-ViT	0.75	80.7(+0.9)	2.6(↓ 43%)	3701(↑1.32×)
CF-ViT	1.0	80.8(+1.0)	4.0(↓ 13%)	2760(↑1.06×)
LV-ViT-S	-	83.3	6.6	1681
CF-ViT	0.63	83.3(+0.0)	3.1(↓ 53%)	3393(↑2.01×)
CF-ViT	0.75	83.5(+0.2)	4.0(↓ 39%)	2827(↑1.68×)
CF-ViT	1.0	83.6(+0.3)	6.1(↓ 7%)	2022(↑1.31×)

Table 2: Comparison between CF-ViT and its backbones.

### Training Strategy

During the training of our CV-ViT, we set the confidence threshold  $\eta = 1$ , which means the fine inference stage will be always executed for every input image. On the one hand, we expect the fine-grained splitting can well fit the ground truth label  $\mathbf{y}$  for an accurate prediction of the input. On the other hand, we expect the coarse-grained splitting obsesses a similar output with that of fine-grained splitting such that most input can be well recognized at coarse inference stage, which indicates less computational cost. Consequently, the training loss of our CF-ViT is given below:

$$loss = CE(\mathbf{p}_f, \mathbf{y}) + KL(\mathbf{p}_c, \mathbf{p}_f), \quad (14)$$

where  $CE(\cdot, \cdot)$  and  $KL(\cdot, \cdot)$  respectively represent the cross entropy loss and Kullback-Leibler divergence.

During the inference of our CV-ViT, by varying the value of  $\eta$ , we can obtain a trade-off between computational budget and accuracy performance. A large  $\eta$  means more inputs will be sent to fine inference stage, which indicates better performance but more computational cost, and vice versa.

## Experiments

### Implementation Details

For ease of comparison, following existing studies on ViT (Rao et al. 2021; Liang et al. 2022; Tang et al. 2022b; Xu et al. 2022), we build our CF-ViT with DeiT-S (w/o distillation) (Touvron et al. 2021a) and LV-ViT-S (Jiang et al. 2021) as backbone networks, all of which split each image into  $14 \times 14 = 196$  patches. To show the advantages of our CF-ViT, we conduct the experiments on ImageNet (Deng et al. 2009) from two perspectives: (1) Each input image is split into  $7 \times 7$  ( $N_c = 49$ ) patches at coarse inference stage, leading to a total of  $N_f = 124$  patches at fine inference stage according to Eq. (10). As results, the computation of our CF-ViT is much cheaper than its backbones due to its less split patches. (2) The image patches number at coarse inference stage is changed to  $9 \times 9$ , leading to 204 patches at fine inference stage, which maintains similar FLOPs with the backbones. For a clearer presentation, CF-ViT is denoted as CF-ViT\* in this case.

All training settings of our CF-ViT, such as image processing, learning rate, *etc.*, are to follow these of DeiT and LV-ViT. In the training phase, only conducting the fine-grained splitting at informative regions would affect the con-



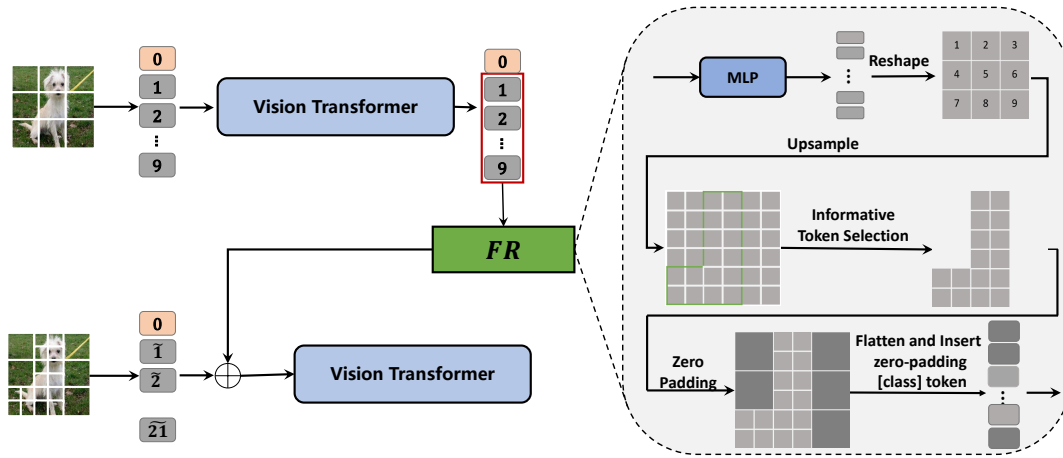


Figure 4: Illustration of our feature reuse module. An MLP is firstly introduced for a flexible transformation among token sequences from coarse stage. Then, four copies of each token are executed as an upsampling strategy and these areas corresponding to fine-grained splitting patches are selected to shortcut to the fine-grained MLP token sequence.

vergence. Therefore, we split the entire image into fine-grained patches in the first 200 epochs, and select informative coarse patches for fine-grained splitting in the remaining training process. Our CF-ViT model is trained on a workstation with 4 A100 GPUs. Notably, both coarse stage and fine stage share the same network parameters. Due to different sizes of patches between coarse stage and fine stage, we downsample the coarse patches to the shape of fine-grained patches in order to facilitate sharing parameters in the patch embedding layer.

## Experimental Results

**Model Efficiency.** To demonstrate our model efficiency, we conduct comparisons between our CF-ViT and its backbones. The measurement metrics include top-1 classification accuracy, model FLOPs and model throughput. Following existing studies (Wang et al. 2021c; Liang et al. 2022), the model throughput is measured as the number of processed images per second on a single A100 GPU. We feed the model 50,000 images in the validation set of ImageNet with a batch size of 1,024, and record the total inference time. Then, the throughput is computed as  $\frac{50,000}{\text{total inference time}}$ .

Table 2 displays the comparison results with varying values of threshold  $\eta$  which balances accuracy and efficiency as discussed in Sec. . It can be observed that when maintaining the same accuracy with the backbone, CF-ViT significantly reduces model FLOPs of DeiT-S by 61% and LV-ViT-S by 53%. Consequently, our CF-ViT obtains a great power to process images, leading to  $1.88\times$  throughput improvements over DeiT-S and  $2.01\times$  over LV-ViT-S. The supreme efficiency is attributed to our design of informative region identification which further splits only informative patches in the coarse stage. Besides, with a larger  $\eta$ , our CF-ViT manifests not only FLOPs and throughput, but better top-1 accuracy. When  $\eta = 1$  which indicates all the input are sent to fine inference stage, our CF-ViT significantly increases the performance of DeiT-S by 1.0% and LV-ViT-S by 0.3%. These

Model	Top-1 Acc.(%)	FLOPs(G)
DeiT-S		
Baseline (Touvron et al. 2021a)	79.8	4.6
DynamicViT (Rao et al. 2021)	79.3	2.9
IA-RED <sup>2</sup> (Pan et al. 2021a)	79.1	3.2
PS-ViT (Tang et al. 2022b)	79.4	2.6
EVIT (Liang et al. 2022)	79.5	3.0
Evo-ViT (Xu et al. 2022)	79.4	3.0
<b>CF-ViT(<math>\eta = 0.5</math>)(Ours)</b>	<b>79.8</b>	<b>1.8</b>
<b>CF-ViT(<math>\eta = 0.75</math>)(Ours)</b>	<b>80.7</b>	<b>2.6</b>
LV-ViT-S		
Baseline (Jiang et al. 2021)	83.3	6.6
DynamicViT (Rao et al. 2021)	83.0	4.6
EVIT (Liang et al. 2022)	83.0	4.7
SiT (Zong et al. 2022)	83.2	4.0
<b>CF-ViT(<math>\eta = 0.63</math>)(Ours)</b>	<b>83.3</b>	<b>3.1</b>
<b>CF-ViT(<math>\eta = 0.75</math>)(Ours)</b>	<b>83.5</b>	<b>4.0</b>

Table 3: Comparisons between existing token slimming based ViT compression methods and our CF-ViT.

results well demonstrate that our CF-ViT can well maintain a trade-off between model performance and model efficiency.

**Comparison with Compressed Models.** To demonstrate the efficacy of our coarse-to-fine patch splitting in reducing model complexity, we further compare our CF-ViT with recent studies on compressing ViT models, including token slimming compression and early-exiting compression.

(1) Token slimming compression reduces the complexity of ViT models by reducing the number of input tokens (patches), which is also the focus of this paper. Table 3 shows the comparison with existing token slimming based ViT compression methods, including DynamicViT (Rao et al. 2021), IA-RED<sup>2</sup> (Pan et al. 2021a), PS-ViT (Tang et al. 2022b), EVIT (Liang et al. 2022), Evo-ViT (Xu et al. 2022) and SiT (Zong et al. 2022). We report top-1 accuracy and FLOPs for performance evaluation. Results with DeiT-S and LV-ViT-S as backbones indicate that our CF-

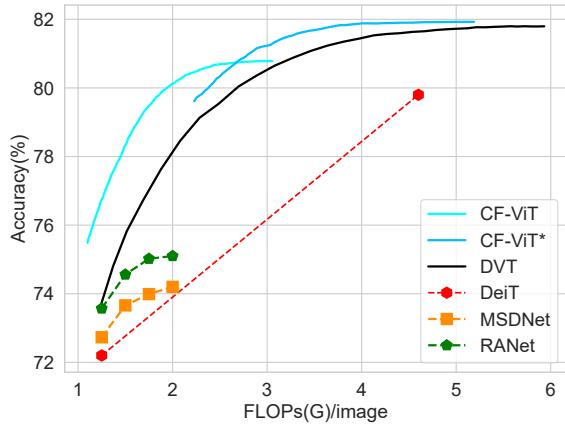


Figure 5: Comparison between our CF-ViT and existing early-exiting methods. MSDNet (Huang et al. 2018) and RANet (Yang et al. 2020) are CNN-based models. DVT (Wang et al. 2021c) and CF-ViT are built upon DeiT.

ViT outperforms previous methods *w.r.t.* accuracy performance and FLOPs reduction. For example, CF-ViT significantly reduces the FLOPs of DeiT-S to 1.8G FLOPs without any compromise on accuracy performance, while the performance of recent advance, Evo-ViT, has only 79.4% with much heavy FLOPs burden of 3.0G. Similar results can be observed when using LV-ViT-S as the backbone.

(2) Early-exiting compression stops the inference if the intermediate representation of an input satisfies a particular criterion, which is also considered in the coarse inference stage of our CF-ViT where the computational graph stops if the prediction confidence  $p_j^c$  exceeds the threshold  $\eta$ . In Fig. 5, we further compare with the early-exiting methods, including CNN-based models such as MSDNet (Huang et al. 2018) and RANet (Yang et al. 2020), as well as transformer-based models such as DVT (Wang et al. 2021c). For fair comparison, both DVT and our CF-ViT are constructed upon DeiT-S as the backbone. From Fig. 5, two phenomena can be observed: (1) Transformer-based models usually show supreme performance over CNN-based methods under similar FLOPs consumption. (2) Our CF-ViT consistently results in best accuracy than DVT that cascades multiple ViTs with an increasing token number. In contrast, our CF-ViT implements fine-grained splitting only for the informative regions, leading to an overall reduction in tokens. Thus, with similar accuracy, CF-ViT manifests smaller FLOPs consumption.

**Comparison with SOTAs.** Fig. 6 compares the accuracy and FLOPs trade-off of popular ViT models as well as our CF-ViT built upon LV-ViT-S (Jiang et al. 2021). The compared methods include DeiT (Touvron et al. 2021a), PVT (Wang et al. 2021a), CoaT (Xu et al. 2021), CrossViT (Chen, Fan, and Panda 2021), Swin (Liu et al. 2021), T2T-ViT (Yuan et al. 2021b), CaiT (Touvron et al. 2021b), iFormers (Si et al. 2022), and EfficientNet (Tan and Le 2019). It can be observed that CF-ViT is significantly competitive in computation-accuracy trade-off among these

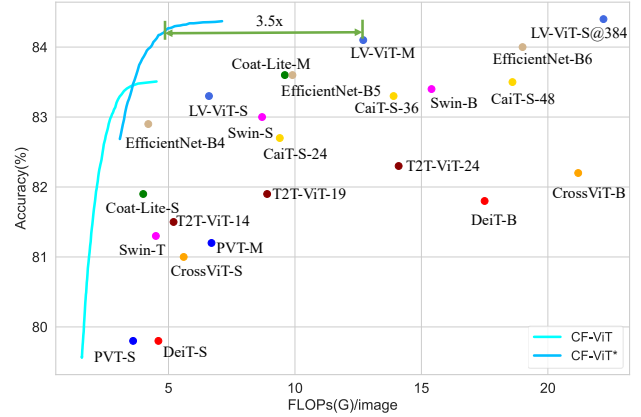


Figure 6: Comparison with popular ViT models. Our CF-ViT is built upon LV-ViT-S.

Ablation	Top-1 Acc.(%)	
	coarse	fine
negative GCA	74.9	77.6
random	75.3	79.6
last class attention	75.3	80.3
<b>Ours</b>	<b>75.5</b>	<b>80.8</b>

Table 4: Performance comparison between our informative region identification and its variants. GCA means global class attention.

baselines. For example, CF-ViT\* achieves 84.1% accuracy with  $3.5\times$  less FLOPs compared with the vanilla LV-ViT-M.

### Ablation Study

We analyze the efficacy of each design in our CF-ViT, including informative region identification, feature reuse and early-exiting. To show the effectiveness of our designs, we also compare our informative region identification and feature reuse with other alternatives. To show their necessity, we remove each design individually and display the performance. All ablation studies take DeiT-S as backbone.

**Informative region identification.** Three variants are developed to replace our informative region identification: (1) Negative global class attention, which selects the regions with smaller global class attention. (2) Random, which randomly picks up regions for fine-grained. (3) Last class attention, which leverages the class attention in the last encoder. We deactivate the early termination, and compare the performance in Table 4. It is intuitive that the negative global class attention results in the poorest performance since the most informative are removed in this setting. This result is in coincidence with the observation in Fig. 1(a). Our informative region identification considers the global class attention, leading to performance increase compared to this only considering the last class attention, well demonstrating the correctness of our motive to combine class attention across different layers.

**Feature Reuse.** Our feature reuse leverages a MLP to pro-

Ablation	Top-1 Acc.(%)	
	coarse	fine
w/o reuse	75.2	80.0
Ours + [class] token	75.4	80.2
Ours + uninformative tokens	75.4	80.6
MLP→Linear	75.3	80.6
<b>Ours</b>	<b>75.5</b>	<b>80.8</b>

Table 5: Performance comparison between our feature reuse and its variants.

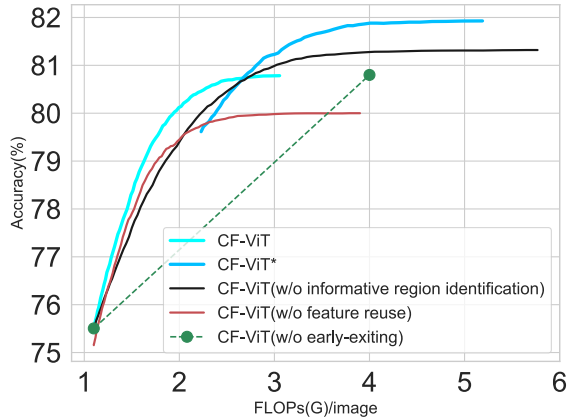


Figure 7: Performance analysis of removing each of the three designs.

	0.4	<b>0.5(default)</b>	0.6	0.7	0.8	0.9
Top1-Acc(%)	80.4	80.8	80.9	81.1	81.3	81.4
FLOPs(G)	3.7	4.0	4.4	4.7	5.1	5.5

Table 6: Accuracy and FLOPs with different values of  $\alpha$ .

	0	0.5	0.9	<b>0.99(default)</b>	0.999
Top1-Acc(%)	80.3	80.5	80.7	80.8	80.8

Table 7: Accuracy with different values of  $\beta$ .

cess the output image token in the coarse inference stage and shortcuts them to the fine inference stage. Note that we do not reuse the [class] token and uninformative tokens. Table 5 compares our feature reuse with three variants including: (1) Integrating [class] token to our feature reuse. (2) Integrating uninformative tokens to our feature reuse. (3) Replacing the MLP with one single linear layer. From Table 5, we can see that considering [class] token or uninformative tokens has a negative impact on the performance. Besides, replacing the MLP layer with a linear layer drops down the performance from 80.8% to 80.6%. These results well demonstrate the effectiveness of our design of feature reuse.

**Necessity of each design.** Fig. 7 plots the performance of our CF-ViT by individually removing each design. Generally, we can observe that the removal of each component incurs severe performance drops. Thus, all three designs are vital to the final performance of our CF-ViT. It is worth

Ablation	Top-1 Acc.(%)	
	coarse	fine
CE + CE	<b>75.7</b>	80.3
<b>CE + KL(ours)</b>	75.5	<b>80.8</b>

Table 8: Performance comparison between different loss function.

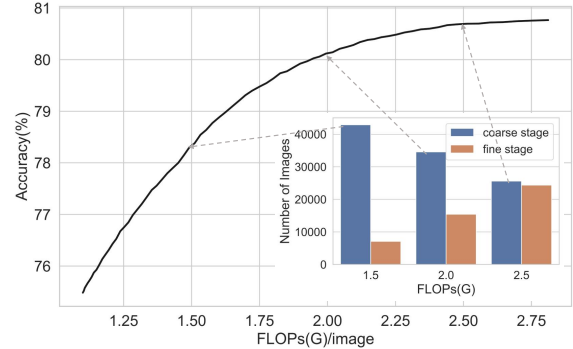


Figure 8: No. of images correctly classified at coarse and fine stages.

noting that, the one without informative region identification split all the coarse-grained patches, leading to a drastic increase of tokens. However, our informative identification chooses to split only informative regions, which greatly reduces tokens and thus brings less computational cost.

**Influence of  $\alpha$ .** Tab. 6 provides accuracy of fine inference stage and FLOPs with different values of  $\alpha$  (see Eq. 10). We can see that from Tab. 6 that, a larger  $\alpha$  leads to better accuracy but more FLOPs consumption. In this paper, we set  $\alpha$  as 0.5 for an accuracy-FLOPs trade-off.

**Influence of  $\beta$ .** Tab. 7 provides accuracy of fine inference stage with different values of  $\beta$  (see Eq. 9). It is intuitive that  $\beta$  indicates the weight of attention from the shallow encoder. The  $\beta = 0$  indicates that only the class attention from the last encoder is used. We set  $\beta = 0.99$  as default for its optimal performance.

**Influence of loss function.** As shown in Eq. (14), we use  $CE(\cdot, \cdot)$  to make the output of fine stage fit the truth label, and use  $KL(\cdot, \cdot)$  to make the output of coarse stage fit the output of fine stage. We also try to make the output of coarse stage and the output of fine stage both fit the truth label:

$$\hat{loss} = CE(\mathbf{p}_f, \mathbf{y}) + CE(\mathbf{p}_c, \mathbf{y}), \quad (15)$$

As show in Tab. 8, compare with origin CE+KL (Eq. (14)), CE+CE (Eq. (15)) implement cause slightly benefit (+0.2%) in coarse inference stage but more degradation (-0.5%) in fine inference stage. We choose Eq. (14) as the loss function because of the significant benefits in fine inference stage.

## Visualization

In Fig. 9, we illustrate some images that are correctly recognized at coarse inference by CF-ViT(DeiT-S), as well as



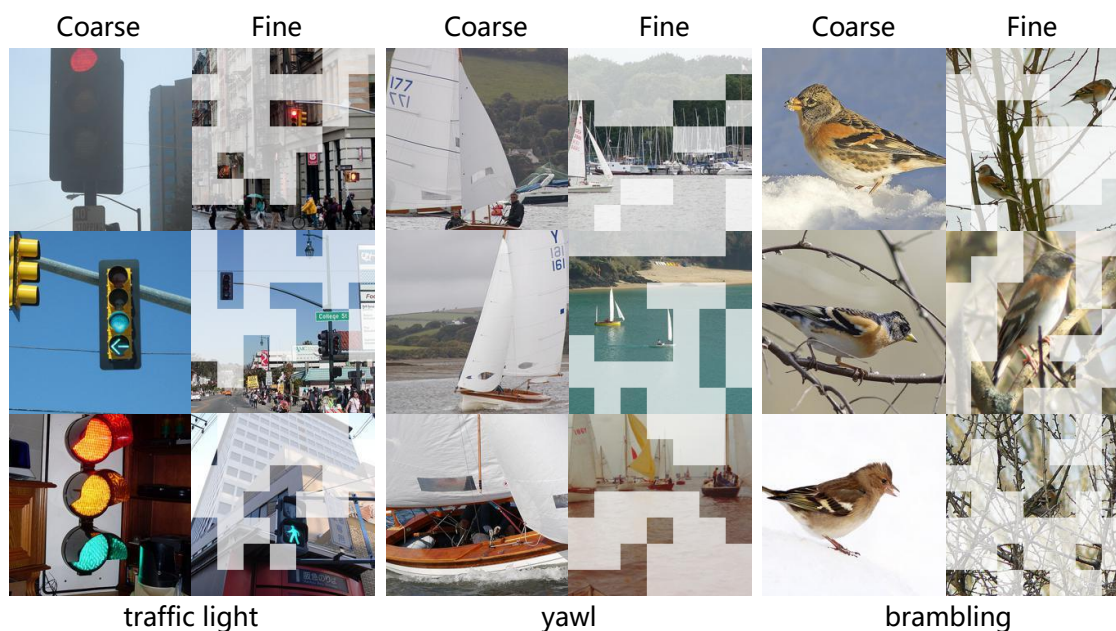


Figure 9: Illustration of images correctly classified at coarse stage and fine stage. For fine stage, we visualize the regions selected by our informative region identification (grey boxes) indicate the uninformative patches.

some recognized at fine inference stage. For a better illustration, we only visualize informative regions if images are recognized at fine inference stage. We can observe an overall trend that images well classified at coarse stage are mostly filled with “easy” regions. Consequently, a coarse-grained splitting can well tell the categories of these images. On the contrary, these containing complex scenes and obscure objects require to be further split for a correct recognition, which is realized at fine stage. Besides, the selected regions by our informative region identification mostly locate the target objects. In Fig. 8, we also show some statistics *w.r.t.* the number of images correctly classified at coarse stage and fine stage. By adjusting the threshold  $\eta$ , CF-ViT models can be obtained with different computational budgets. With a larger  $\eta$ , more images will be further split and fed to fine inference stage for recognition. Therefore, the increasing  $\eta$  brings about more computational cost and the number of images correctly classified at fine stage also increases.

## Conclusion

This paper focuses on reducing the redundant input tokens for accelerating vision transformers. Specially, we proposed a coarse-to-fine vision transformer (CF-ViT), the inference of which is two-fold including a coarse inference stage and a fine inference stage. The former splits the input image into a small-length token sequence to recognize these images filled with “easy” regions in a computationally economical manner while the latter further splits the informative patches for a better recognition if the coarse inference does not well classify the input. Extensive experiments indicate that our CF-ViT can achieve a better trade-off between performance and efficiency. Furthermore, transferring the coarse-to-fine

inference paradigm to dense prediction tasks such as object detection and semantic segmentation will be included in our future work.

## Acknowledgements

This work was supported by the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No. U21B2037, No. U22B2051, No. 62176222, No. 62176223, No. 62176226, No. 62072386, No. 62072387, No. 62072389, No. 62002305 and No. 62272401), Guangdong Basic and Applied Basic Research Foundation(No.2019B1515120049), and the Natural Science Foundation of Fujian Province of China (No.2021J01002, No.2022J06001).

## References

- Chen, C.-F. R.; Fan, Q.; and Panda, R. 2021. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 357–366.
- Chu, X.; Tian, Z.; Wang, Y.; Zhang, B.; Ren, H.; Wei, X.; Xia, H.; and Shen, C. 2021a. Twins: Revisiting the design of spatial attention in vision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Chu, X.; Tian, Z.; Zhang, B.; Wang, X.; Wei, X.; Xia, H.; and Shen, C. 2021b. Conditional positional encodings for vision transformers. arXiv:2102.10882.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 248–255.

- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*.
- Fang, J.; Xie, L.; Wang, X.; Zhang, X.; Liu, W.; and Tian, Q. 2021. Msg-transformer: Exchanging local spatial information by manipulating messenger tokens. arXiv:2105.15168.
- Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. 2022a. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.
- Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; and Wang, Y. 2021a. Transformer in transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Han, Y.; Huang, G.; Song, S.; Yang, L.; Wang, H.; and Wang, Y. 2021b. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.
- Han, Y.; Huang, G.; Song, S.; Yang, L.; Zhang, Y.; and Jiang, H. 2021c. Spatially adaptive feature refinement for efficient inference. *IEEE Transactions on Image Processing*, 9345–9358.
- Han, Y.; Yuan, Z.; Pu, Y.; Xue, C.; Song, S.; Sun, G.; and Huang, G. 2022b. Latency-aware Spatial-wise Dynamic Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Heo, B.; Yun, S.; Han, D.; Chun, S.; Choe, J.; and Oh, S. J. 2021. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 11936–11945.
- Huang, G.; Chen, D.; Li, T.; Wu, F.; van der Maaten, L.; and Weinberger, K. Q. 2018. Multi-scale dense networks for resource efficient image classification. In *International Conference on Learning Representations (ICLR)*.
- Huang, Z.; Ben, Y.; Luo, G.; Cheng, P.; Yu, G.; and Fu, B. 2021. Shuffle transformer: Rethinking spatial shuffle for vision transformer. arXiv:2106.03650.
- Jiang, Z.-H.; Hou, Q.; Yuan, L.; Zhou, D.; Shi, Y.; Jin, X.; Wang, A.; and Feng, J. 2021. All tokens matter: Token labeling for training better vision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Li, S.; Wang, Z.; Liu, Z.; Tan, C.; Lin, H.; Wu, D.; Chen, Z.; Zheng, J.; and Li, S. Z. 2022. Efficient Multi-order Gated Aggregation Network. arXiv:2211.03295.
- Li, Y.; Zhang, K.; Cao, J.; Timofte, R.; and Van Gool, L. 2021. Localvit: Bringing locality to vision transformers. arXiv:2104.05707.
- Liang, Y.; GE, C.; Tong, Z.; Song, Y.; Wang, J.; and Xie, P. 2022. EviT: Expediting Vision Transformers via Token Reorganizations. In *International Conference on Learning Representations (ICLR)*.
- Lin, M.; Chen, M.; Zhang, Y.; Li, K.; Shen, Y.; Shen, C.; and Ji, R. 2022. Super Vision Transformer. arXiv:2205.11397.
- Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10012–10022.
- Pan, B.; Panda, R.; Jiang, Y.; Wang, Z.; Feris, R.; and Oliva, A. 2021a. IA-RED<sup>2</sup>: Interpretability-Aware Redundancy Reduction for Vision Transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Pan, Z.; Zhuang, B.; Liu, J.; He, H.; and Cai, J. 2021b. Scalable vision transformers with hierarchical pooling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 377–386.
- Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; and Hsieh, C.-J. 2021. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Si, C.; Yu, W.; Zhou, P.; Zhou, Y.; Wang, X.; and Yan, S. 2022. Inception Transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Song, L.; Zhang, S.; Liu, S.; Li, Z.; He, X.; Sun, H.; Sun, J.; and Zheng, N. 2021. Dynamic grained encoder for vision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Sun, C.; Shrivastava, A.; Singh, S.; and Gupta, A. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 843–852.
- Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 6105–6114.
- Tang, S.; Zhang, J.; Zhu, S.; and Tan, P. 2022a. Quadtree Attention for Vision Transformers. In *International Conference on Learning Representations (ICLR)*.
- Tang, Y.; Han, K.; Wang, Y.; Xu, C.; Guo, J.; Xu, C.; and Tao, D. 2022b. Patch slimming for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12165–12174.
- Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021a. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, 10347–10357.
- Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; and Jégou, H. 2021b. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 32–42.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Wang, W.; Stuijk, S.; and De Haan, G. 2014. Exploiting spatial redundancy of image sensor for motion robust rPPG. *IEEE Transactions on Biomedical Engineering (TBE)*, 415–425.

- Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021a. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 568–578.
- Wang, Y.; Chen, Z.; Jiang, H.; Song, S.; Han, Y.; and Huang, G. 2021b. Adaptive focus for efficient video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 16249–16258.
- Wang, Y.; Huang, R.; Song, S.; Huang, Z.; and Huang, G. 2021c. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Wang, Y.; Lv, K.; Huang, R.; Song, S.; Yang, L.; and Huang, G. 2020. Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2432–2444.
- Wang, Y.; Yue, Y.; Lin, Y.; Jiang, H.; Lai, Z.; Kulikov, V.; Orlov, N.; Shi, H.; and Huang, G. 2022a. Adafocus v2: End-to-end training of spatial dynamic networks for video recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20030–20040.
- Wang, Y.; Yue, Y.; Xu, X.; Hassani, A.; Kulikov, V.; Orlov, N.; Song, S.; Shi, H.; and Huang, G. 2022b. AdaFocusV3: On Unified Spatial-Temporal Dynamic Video Recognition. In *European Conference on Computer Vision (ECCV)*, 226–243.
- Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Xu, W.; Xu, Y.; Chang, T.; and Tu, Z. 2021. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9981–9990.
- Xu, Y.; Zhang, Z.; Zhang, M.; Sheng, K.; Li, K.; Dong, W.; Zhang, L.; Xu, C.; and Sun, X. 2022. Evo-ViT: Slow-Fast Token Evolution for Dynamic Vision Transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
- Yang, L.; Han, Y.; Chen, X.; Song, S.; Dai, J.; and Huang, G. 2020. Resolution adaptive networks for efficient inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2369–2378.
- Yu, Q.; Xia, Y.; Bai, Y.; Lu, Y.; Yuille, A. L.; and Shen, W. 2021. Glance-and-gaze vision transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Yuan, K.; Guo, S.; Liu, Z.; Zhou, A.; Yu, F.; and Wu, W. 2021a. Incorporating convolution designs into visual transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 579–588.
- Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.-H.; Tay, F. E.; Feng, J.; and Yan, S. 2021b. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 558–567.
- Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H.; et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6881–6890.
- Zong, Z.; Li, K.; Song, G.; Wang, Y.; Qiao, Y.; Leng, B.; and Liu, Y. 2022. Self-slimmed vision transformer. In *European Conference on Computer Vision (ECCV)*, 432–448.