Adaptive split-fusion transformer

Zixuan SU

Jingjing CHEN

Lei PANG

Chong-wah NGO
Singapore Management University, cwngo@smu.edu.sg

Yu-Gang JIANG

Citation

Adaptive Split-Fusion Transformer

Zixuan Su, Hao Zhang, Jingjing Chen, *Member, IEEE*, Lei Pang, Chong-Wah Ngo, *Member, IEEE*
and Yu-Gang Jiang, *Senior Member, IEEE*

Abstract—Neural networks for visual content understanding have recently evolved from convolutional ones to transformers. The prior (CNN) relies on small-windowed kernels to capture the regional clues, demonstrating solid local expressiveness. On the contrary, the latter (transformer) establishes long-range global connections between localities for holistic learning. Inspired by this complementary nature, there is a growing interest in designing hybrid models which utilize both techniques. Current hybrids merely replace convolutions as simple approximations of linear projection or juxtapose a convolution branch with attention without considering the importance of local/global modeling. To tackle this, we propose a new hybrid named *Adaptive Split-Fusion Transformer* (ASF-former) that treats convolutional and attention branches differently with adaptive weights. Specifically, an ASF-former encoder equally splits feature channels into half to fit dual-path inputs. Then, the outputs of the dual-path are fused with weights calculated from visual cues. We also design a compact convolutional path from a concern of efficiency. Extensive experiments on standard benchmarks, such as ImageNet-1K, CIFAR-10, and CIFAR-100, show that our ASF-former outperforms its CNN, transformer, and hybrid counterparts in terms of accuracy (83.9% on ImageNet-1K), under similar conditions (12.9G MACs/56.7M Params, without large-scale pre-training). The code is available at: https://github.com/szx503045266/ASF-former.

Index Terms—Visual understanding, Transformer, CNN, Hybrid, Gating.

I. Introduction

Neural networks for learning visual representations have recently split into directions of conventional convolutional neural networks (i.e., CNN) and emerging transformers. CNN used to be the de-facto standard network and was good at modeling localities. On the contrary, the transformer learns holistic features by building pair-wise relations, thus demonstrating strong global expressiveness. Pilot visual transformers, including ViT [2], T2T-ViT [1], deliberately avoid convolutions and only rely on the self-attention. Though achieving good accuracy, they pay extra computations as a price for bypassing efficient convolution operators.

Since convolutions and self-attention are complementary when viewed from perspectives like local-global modeling and high-low efficiency, it is natural to study hybrid networks to ensure each part serves its best. Existing hybrids usually combine these two parts in a cascade or parallel manner. For a cascade hybrid, researchers usually re-implement linear projections in vanilla transformers with convolutional

Zixuan Su, Jingjing Chen and Yu-Gang Jiang are with School of Computer Science, Fudan University, Shanghai, China (E-mail: zxsu21@m.fudan.edu.cn, {chenjingjing,ygj}@fudan.edu.cn). Hao Zhang and Chong-Wah Ngo are with Singapore Management University, Singapore (E-mail: {hzhang,cwngo}@smu.edu.sg). Lei Pang is with City University of Hong Kong, Hong Kong, China (E-mail: cactuslei@gmail.com).

The Corresponding author is Jingjing Chen.

approximations. For example, token-embedding [3]–[9] and linear-projections [3], [5]–[7], [9], [10] in attentions/MLPs are commonly replaced by convolutions. These cascade works share a common principle of minimal change. As for parallel hybrids, an extra convolutional branch is inserted on par with the attention in a dual-branch (or dual-path) manner [11]–[14]. This strategy enables learning local/global visual contexts independently and is beneficial for analyzing the effectiveness of each path.

However, most current hybrid models treat local and global contexts equally, which conflicts with the real-world scenario that the importance of local/global cues varies according to the image visuals and layer depth. For example, tiny objects prefer local evidence, whereas landscapes bias global views in the recognition process. Besides, layer with different depths also shows their biases in learning local/global contexts, as mentioned in [12].

To tackle this, we propose a novel parallel hybrid named *Adaptive Split-Fusion Transformer* (ASF-former), which adopts an adaptive gating strategy to select convolution/attention paths according to global visual cues. Its encoder contains two parts: *Efficient Split Parallelism with HMCB* and *Adaptive Fusion* as shown in Figure (1).

Efficient Split Parallelism with HMCB differs from the existing parallel hybrid models in two aspects. Firstly, we comprehensively and carefully craft an efficient convolution path named *Half-Residual Mobile Convolutional Branch* (HMCB). Compared with existing counterparts, the HMCB demonstrates stronger local capability with fewer computations. Secondly, we split inherent feature channels of pure transformers into half and separately feed sub-features into the Conv/attention branch. Thereby, we could keep the same feature dimension as the original backbone. With these, the Split Parallelism shares a similar complexity as single-path (convolution or attention) models.

Adaptive Fusion module intakes outputs from convolution and attention branches and weighs them with adaptive scalars. Specifically, visual features from both paths are processed by sequential layers, including global pooling, fully-connected layer, and Sigmoid activation, to generate weighting scalars. These scalars are used to weigh features from each branch (Fig. (1)). We also add an extra skip connection to alleviate gradient vanishing in backpropagation. We experimentally verify that the new adaptive fusion could effectively and efficiently select convolution/attention branches according to visual contents. We briefly summarize our contributions below.

- *Efficient Split Parallelism with HMCB*. We introduce a new Half-Residual Mobile Convolutional Branch,
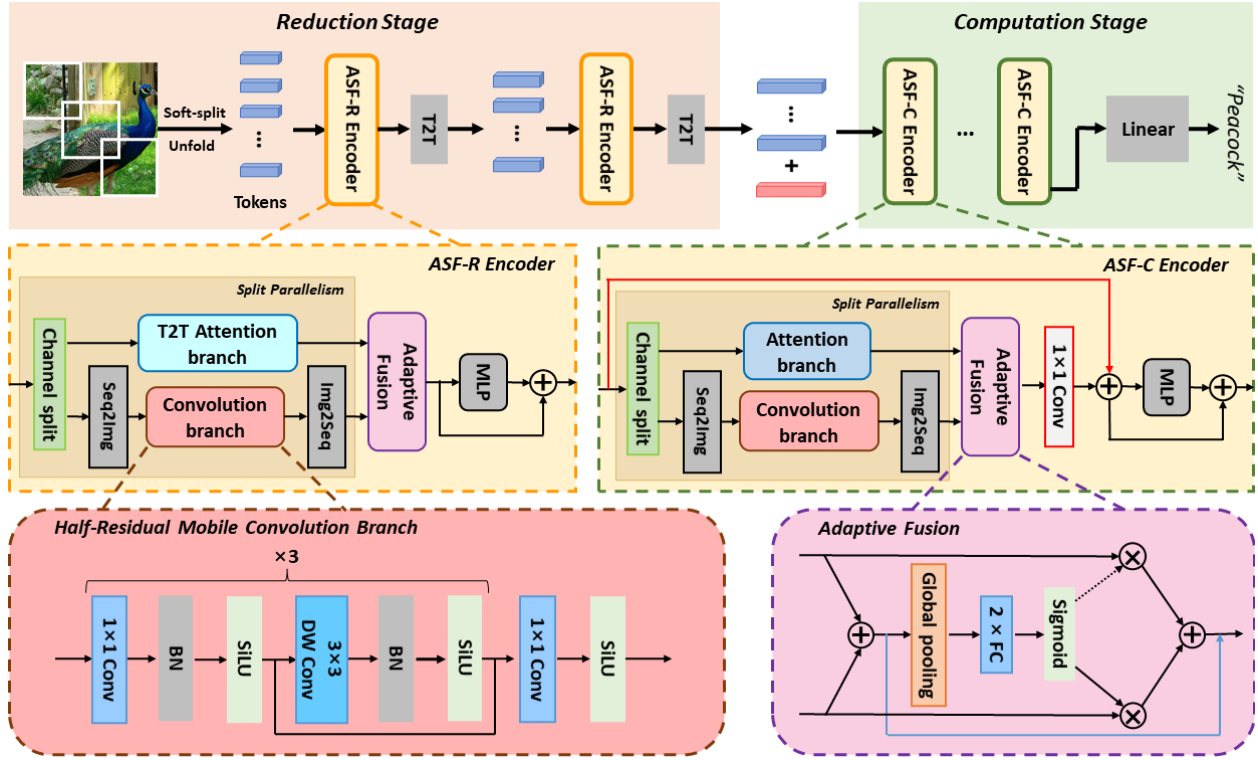
Fig. 1. An overview of the ASF-former. The encoders in a reduction and computation stage are separately denoted as ASR-R/C. Both types include Split Parallelism and Adaptive Fusion parts, except that the ASF-R adopts T2T attention for down-sampling token dimension. As shortcut and Conv 1×1(red line) is incompatible with the T2T attention [1], they are removed in the reduction stage. (This figure is best viewed in color)

which complements the attention feature with high cost-effectiveness.

- *Adaptive Fusion.* We first introduce adaptive fusion to combine outputs from convolution and attention branches. We could adjust the importance of local/global modeling with adaptive weights according to visual contents.
- *A new hybrid ASF-former.* We build a new CNN-transformer hybrid with efficient convolution layers (HMCB) and an adaptive fusion strategy. Experiments on standard benchmarks, such as ImageNet-1K and down-stream datasets, show that the ASF-former could achieve SOTA performance (83.9% on ImageNet-1K) under similar conditions (12.9G MACs / 56.7M Params).

II. RELATED WORK

Convolutional Neural Network. Since the birth of AlexNet [15], convolutional neural network (CNN) was the de-facto standard model on different vision topics, such as image recognition [16]–[21], object detection [22]–[28], instance/semantic segmentation [29]–[31] and video recognition [32]–[38].

To apply CNN to edge devices, researchers spare many efforts on balancing complexity and capacity. For example, MobileNet [39] factorized a standard convolution with depthwise separable convolution. Moreover, MobileNetV2 [40] introduced an extra inverted residual design and expanded the usage of depthwise convolution into more layers (e.g., intermedia expansion layer). ShuffleNet [41] combines group convolution, channel shuffle, and depthwise convolution to build an efficient

CNN unit. EfficientNet [42] follows an empirical scaling rule to search and adjust hyperparameters of network structure, such as "*dimension, depth, width, resolution*", to achieve a high cost-effective.

Some researchers proposed plug-and-play modules introducing a few extra parameters to recalibrate CNN features. For example, the SE-Net [43] introduced a new "Squeeze-and-Excitation" block on top of a plain 2D-CNN backbone. The block could adaptively adjust feature channels with global contexts. Furthermore, GE-Net [44] studied more choices of "Gathering" contexts information than SE-Net. Hao et al. [45] extended contextual calibrating for videos and proposed GC-Net that utilizes spatial-temporal contexts. SKNet [46] selectively fuse two convolution branches with different kernel sizes to adaptively adjust the receptive field size. ResNeSt [47] further combined the channel-wise attention with a multi-path strategy. This design would build cross-feature interactions and benefit learning diverse representations.

Vision Transformer. Transformer receives extensive interest in various multimedia and vision tasks, such as Video Recognition [48], Image Understanding [49], [50], Re-Identification [51]–[53], Visual Inpainting [54], Pose Estimation [55] tasks since the birth of ViT [2], which validates the feasibility of replacing CNNs with pure transformers with large-scale pre-training.

Though achieving impressive accuracy, the ViT [2] suffers from a high computational cost. The cost is caused by densely calculating the pair-wise distance between visual tokens in each attention module. To balance computations and classi-

fication accuracy, researchers spare more effort on developing new transformers, including DeiT [56], T2T-ViT [1], TNT [57], PVT [58], Swin [59] than before. Specifically, DeiT [56] adopted convnet as a teacher and trained transformer under a teacher-student strategy. It relies on the distillation token to introduce locality into a transformer, thus lowering the requirement for large-scale training data. T2T-ViT [1] focused on shrinking token-length. It designs the T2T module to down-sampling tokens via concatenating features of local neighboring pixels. TNT [57] altered the granularity of patch dividing by further splitting local patches into subpatches. The inner attention will be calculated within each patch before outer global attention. In order to be ported to various downstream tasks, PVT [58] introduced a progressive shrinking pyramid structure and a spatial-reduction attention for efficiently learning multi-scale and high-resolution features. For further parameter and computation efficiency, Swin Transformer [59] utilized shifted window to split the feature map and performed self-attention within each local window. These models are pure convolutional-free transformers, thus lacking local capacity and convolution's efficiency strength.

Hybrid Transformer. Attracted by the complementary nature of CNN and Attentions, more and more efforts are devoted to developing hybrid transformers. Existing hybrids can be separated into two groups. The first is cascade hybrid which minimally modify the original transformer model by re-implementing the token-embedding [3]–[9] and the linear projections [3], [5]–[7], [9], [10] in Attentions/MLPs with convolution operators. The second is parallel hybrid which juxtaposes an extra convolutional branch on par with the attention [11]–[14]. For example, Conformer [13] designed the Feature Coupling Unit (FCU) for transmitting features from one path to another. For acquiring inductive bias from convolution, ViTAE [14] built the parallel structure in each block and designed the pyramid reduction module with dilated convolution. These methods treat convolution and attention paths equally. ACmix [12] instead set two learnable weights for measuring the importance of two paths, but the weights only vary with network depth, failing to be adjusted according to visual contexts.

III. Our Method

An overview of ASF-former is shown in Figure (1). Similar to [1], [14], it contains a total of $L = L_1 + L_2$ encoders, where L_1/L_2 encoders reside in *reduction* or *computation* stages. As in [1], the two stages differentiate in whether adopting the T2T for shrinking token-length and T2T attentions for reducing computations. To distinguish, we separately denote encoders in the two stages as the ASF-R and ASF-C. We present a detailed pipeline of ASF-former below.

An image $I \in \mathbb{R}^{H \times W \times 3}$ is first soft-split into patches. Each patch shares an identical shape of $k \times k$ with overlap o and padding p. These patches are unfolded into a sequence of tokens $T_0 \in \mathbb{R}^{N_0 \times D_0}$, where $D_0 = 3k^2$, and token-length is:

$$N_0 = \left\lfloor \frac{H + 2p - k}{k - o} + 1 \right\rfloor \times \left\lfloor \frac{W + 2p - k}{k - o} + 1 \right\rfloor. \quad (1)$$

Tokens T_0 go through the two stages, including the reduction and computation stages for representation learning.

Reduction stage contains L_1 replicated ASF-R + T2T pairs, where the prior and the latter module separately serve for feature learning and down-sampling. Denote tokens from the i-th pair as $T_i \in \mathbb{R}^{N_i \times D_i}$ or $\widetilde{T}_i \in \mathbb{R}^{N_i \times D'}$. The token-length N_i and dimension D_i would reduce and increase according to the depth $i \in [1, 2, \cdots, L_1]$, due to the T2T operation, while the ASF-R encoder would decrease the token dimension to D'. A math process of the i-th pair is shown as:

$$\widetilde{T}_{i-1} = f_{ar}(T_{i-1}) \quad (2)$$

$$T_i = f_{t2t}(\widetilde{T}_{i-1}) \quad (3)$$

where $f_{ar}(\cdot)$ and $f_{t2t}(\cdot)$ denotes the ASR-R and T2T modules.

Output $T_{out} \in \mathbb{R}^{N_{L_1} \times D}$ of reduction stage is obtained by linear-projecting T_{L_1} to a fixed D-dimensional space.

$$T_{out} = Linear(T_{L_1}) \quad (4)$$

Computation stage contains L_2 identical ASF-C encoders, without changing token-length. Same as the ViT [2], an extra [CLASS] token $C_0 \in \mathbb{R}^{1 \times D}$ is concatenated with T_{out} for an input $X_0 \in \mathbb{R}^{(N_{L_1}+1) \times D}$ of this stage. Notably, the [CLASS] part would only be processed by the attention branch.

$$X_0 = [T_{out}; C_0] \quad (5)$$

Denoting the ASF-C with function $f_{ac}(\cdot)$, the process of the j-th encoders is:

$$X_j = f_{ac}(X_{j-1}), \quad X_j \in \mathbb{R}^{(N_{L_1}+1) \times D} \quad (6)$$

The [CLASS] token yielded by the last ASF-C encoders will be fed into a fully-connected layer for category prediction:

$$Y = Linear(C_{L_2}), \quad Y \in \mathbb{R}^{Categories} \quad (7)$$

Since ASF-R/C encoders share most parts, we present them together in Section III-A.

A. An ASF-R/C Encoder

The ASF-R & ASF-C encoders are same in *Split Parallelism*, *Adaptive Fusion* and MLP parts, and differs in the attention part (T2T or vanilla attention).

Split Parallelism equally split a tensor of tokens $T \in \mathbb{R}^{N \times D}$ for the ASF-R (or X for the ASF-C) into two parts $T^{(a)}, T^{(b)} \in \mathbb{R}^{N \times \frac{D}{2}}$, along the channel axis. Then, the subtensor $T^{(a)}/T^{(b)}$ is separately fed into convolutional/attention branch for local/global modeling. Notably, $T^{(a)}$ are pre/post-processed with *seq2image* or *image2seq* function [1] to rearrange tokens into spatial or sequential form. The process is shown below:

$$\hat{T}^{(a)} = img2seq\left(f_{convb}\left(seq2img\left(T^{(a)}\right)\right)\right) \quad (8)$$

$$\hat{T}^{(b)} = f_{atteb}\left(T^{(b)}\right) \quad (9)$$

where $f_{atteb}(\cdot)$ and $f_{convb}(\cdot)$ respectively denote attention and convolution paths, and $\hat{T}^{(a)}, \hat{T}^{(b)} \in \mathbb{R}^{N \times D'}$. Hereby, D'=64 in the ASF-R (or $D'=\frac{D}{2}$ in ASF-C). Notably, we carefully
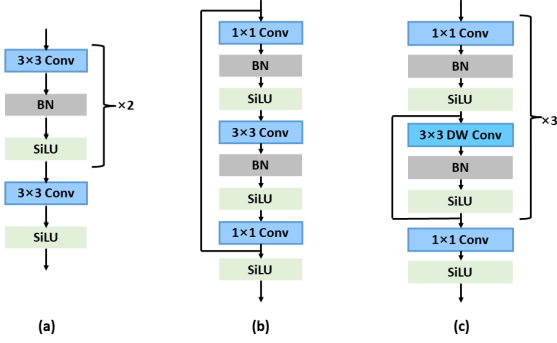
Fig. 2. Designs of convolutional branch: (a) PCM from ViTAE transformer [14]; (b) Residual bottleneck; (c) our **HMCB**
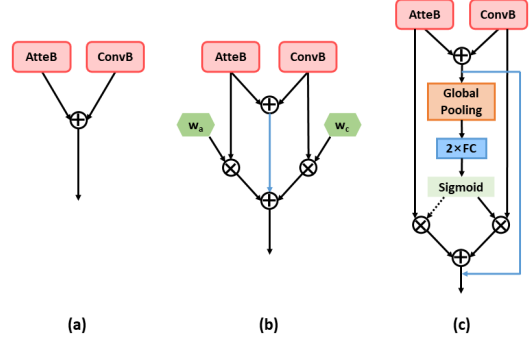

Fig. 3. Fusion strategies: (a) Simple Fusion; (b) Context-Agnostic Fusion; (c) Adaptive Fusion. The shortcut (blue) serves for reducing gradient vanishment

craft an efficient convolutional branch named *Half-Residual Mobile Convolutional Branch* and present it in Section III-B.

Adaptive Fusion performs weighted sum on tensors processed by the two paths with adaptive scalars α and β. Hereby, α and β are calculated according to visual features from the two paths by Eq. (11)$\sim$(12).

$$S = \hat{T}^{(a)} + \hat{T}^{(b)} \tag{10}$$

$$\alpha = Sigmoid\left(f_w\left(S\right)\right) \tag{11}$$

$$\beta = 1 - \alpha \tag{12}$$

$$\hat{T} = \alpha \cdot \hat{T}^{(a)} + \beta \cdot \hat{T}^{(b)} + S \tag{13}$$

where the $f_w\left(\cdot\right)$ denotes the function for generating weighting scalars. Notably, we generate the α & β in a *Sigmoid* way. Though this way is theoretically equivalent to a *Softmax* function, it is practically simple in implementation. We describe details and compare different fusion strategies in Section III-C.

Attentions & MLP are mostly inherited from the general vision transformer regime, with minor modifications on attention settings. Specifically, the ASF-R/C separately adopt the T2T attention and vanilla attentions. Compared with the vanilla, the T2T attention replaces the multi-head scheme with a single-head one and fixes channels of "query", "key", "value" to $D' = 64$, concerning computational efficiency. Since the T2T attention reshapes tokens, the shortcut and Conv 1×1 are removed in the ASF-R compared with the ASF-C (red line in Fig. (1)). Output $\widetilde{T}/\widetilde{X}$ of the ASF-R/C encoders is generated as in Eq. (14)$\sim$(15), where $f_{mlp}\left(\cdot\right)$ denotes the MLP with two *fc* layers and a GeLU activation:

$$\widetilde{T} = f_{mlp}\left(\hat{T}\right) + \hat{T} \tag{14}$$

$$\widetilde{X} = f_{mlp}\left(\mathring{X}\right) + \mathring{X}, \qquad \mathring{X} = Conv\left(\hat{X}\right) + X \tag{15}$$

B. Half-Residual Mobile Convolutional Branch

We study existing CNN branches for hybrid models and craft a new *Half-Residual Mobile Convolutional Branch* (HMCB). The HMCB is more complementary to the attention way than its counterparts while consuming fewer computations. We begin with PCM, the recently proposed CNN-branch in ViTAE hybrid [14].

Parallel Convolution Module (PCM) is shown in Figure (2a), it contains stacked convolutions with 3×3 kernels.

We re-implement it by replacing its internal *group conv* with conventional *conv*. This re-implementation is combined with our Split Parallelism and surpasses the original ViTAE hybrid (Accuracy from $82.0\% \rightarrow 82.2\%$), at the expense of extra parameters.

Residual Bottleneck is widely used in various vision tasks (Fig. (2b)). Thus, we want to adopt it as the convolutional branch in the hybrid models. We only implement one block of residual bottleneck here to maintain similar params/MACs as the original PCM.

Half-Residual Mobile Convolutional Branch (HMCB) is modified based on PCM, with the help of efficient *conv* approximations (Fig. (2c)). Inspired by MobileNet [39] and MobileNetV2 [40], we first factorize each conventional 3×3 *conv* into one 3×3 *depth-wise conv* followed by one 1×1 *conv* and then we add another 1×1 *conv* before the first *depth-wise conv*.

These approximations remarkably reduce computations. Even if we replicate the half-residual block three times, the HMCB still contains similar Params / MACs to single Residual bottleneck. Specifically, we implant the shortcut at a different position with the conventional residual bottleneck to be compatible with the repetition and promote the training across channels. We compare the three designs in terms of accuracy, Params, and MACs in Table III and observe that our HMCB performs the best under all metrics.

C. Adaptive Fusion and Counterparts

In this part, we present Adaptive Fusion and two simple counterparts. We begin with a *simple fusion* with fixed weights, then introduce a fusion strategy *context-agnostic* weights, and finally, give the *Adaptive Fusion* with contextually relevant weights.

Simple Fusion directly averages outputs from the two branches with equal importance as in Eq. (16) and Fig. (3a). The fusion itself is parameter-free and effective. Thus, it is preferred in a pilot parallel hybrid, ViTAE [14].

$$\alpha = \beta = 0.5 \tag{16}$$

Context-Agnostic Fusion explicitly learns α & β on par with training process (Fig. (3b)). To avoid a phenomenon that the gradient vanishment deactivates a particular branch
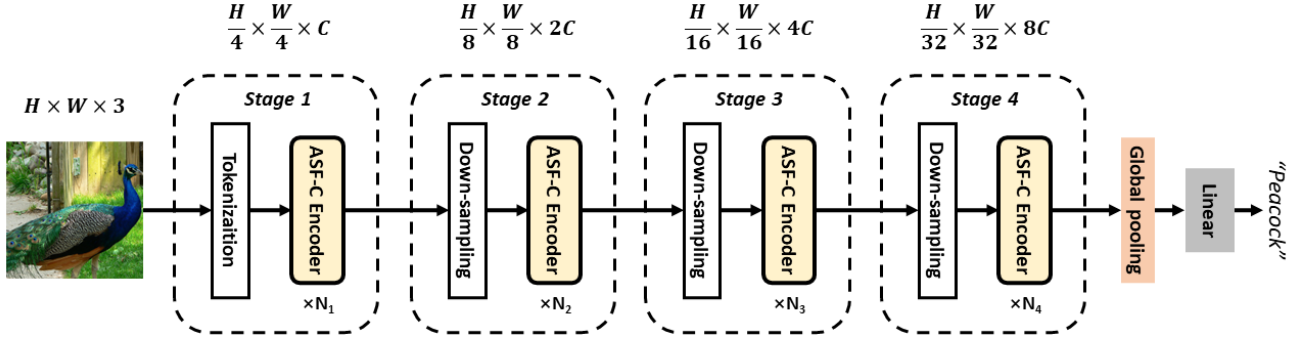
Fig. 4. The architecture of ASF-former$_p$. It adopts a 4-stage pyramid structure and is built based on ASF-C encoders. Without [CLASS] token, the final output feature will be entirely fed into the classification layer after global average pooling.

when W_α or W_β falls into extremely tiny values, we add an extra skip connection (blue line) to enforce gradients to be propagated to both ways.

$$\alpha = W_\alpha, \quad \beta = W_\beta \tag{17}$$

Adaptive Fusion calculates α and β according to visual contexts from both branches, which is inspired by SKNet [46]. Its process is shown in Eq. (18)&(19) and Figure (3c).

$$\alpha = Sigmoid\left(Linear^2\left(Pool\left(\hat{\boldsymbol{T}}^{(a)} + \hat{\boldsymbol{T}}^{(b)}\right)\right)\right) \tag{18}$$

$$\beta = 1 - \alpha \tag{19}$$

Specifically, we expand the function f_w in Eq. (11) to be two fully-connected layers (*Linear*), with *BatchNorm* and *GeLU* activations in between. To stabilize the training procedure, we add the extra skip connection from the same concern as the prior fusion method.

We comprehensively compare various fusion strategies in Table V and observe a significant improvement with our Adaptive Fusion strategy.

D. Pyramid Structure

ASF-former is built beyond the structure of T2T-ViT [1], which has a reduction stage for down-sampling and a computation stage for feature encoding under the same feature dimension. However, downstream tasks such as object detection, instance segmentation, and semantic segmentation usually prefer multi-level features from the backbone model to capture objects at different scales. This requirement makes ASF-former not quite suitable for transferring to downstream tasks. To deal with this problem, we follow the pyramid structure of Swin Transformer [59] and further design a new model with our ASF-C encoder, denoted as ASF-former$_p$.

As Figure (4) shows, ASF-former$_p$ adopts a 4-stage pyramid structure with down-sampling layers before each stage. The original images are tokenized and down-sampled by the ratio of 4 in the first stage, while the other down-sampling layers use a ratio of 2. ASF-former$_p$ has N_1,N_2,N_3,N_4 ASF-C encoders in each stage respectively and the feature channel dimension C in Stage 1 will be doubled in each following stages. Following Swin [59], we use the simple convolutions for image tokenization and down-sampling instead of T2T

operation; therefore, we could remove the ASF-R from the ASF-former$_p$. Besides, instead of adding a [CLASS] token for classification, we perform global average pooling on the output feature of the last stage and directly feed it into a fully-connected layer for category prediction.

We will compare ASF-former and its pyramid version, ASF-former$_p$, in Table VII. Besides, in Section IV-G and IV-H, we will adopt ASF-former$_p$ for all downstream tasks and further utilize its stage-wise nature.

IV. EXPERIMENTS

We evaluate the ASF-former on standard benchmarks, including ImageNet-1K, CIFAR-10/100, with metrics like Top-1/5 accuracy, model Params, and inferencing MACs. Experimental results validate the efficacy and efficiency of the ASF-former.

A. Datasets

We conduct ablation and transferability experiments on ImageNet-1K and CIFAR-10/100 downstream datasets.

ImageNet-1K [60] defines 1000 categories with 1.35 million images captured in daily life. On average, each category contains around 1.3k samples. These images are split into training/validation sets with a ratio of 26:1.

CIFAR-10/100 [61] respectively contains 10/100 categories on 60k images with a fixed 32×32 resolution. In the CIFAR-10, each category includes 6k samples, with 5k/1k samples as training/testing. Whereas, in the CIFAR-100, there are 600 samples per category, with 500/100 for training/testing.

B. Experimental settings

In this part, we briefly introduce hyperparameters of our ASF-former variants and the training & inference recipes.

ASF-former variants. By customizing hyperparameters, such as the number of encoders (i.e., L_1 and L_2) and dimensions of tokens in different layers, we can flexibly control the complexity of ASF-former at different computation scales. To fairly compare the ASF-former with its counterpart of similar computational costs, we propose a *small* and *big* model, respectively denoted as the ASF-former-S and ASF-former-B in Table I. Besides, we set the same k, o, p as the original T2T-ViT model (Eq. (1)). As for ASF-former$_p$,

Model	Reduction stage			Computation stage			Model Size	
	Depth L_1	Token dim D'	MLP dim	Depth L_2	Token dim D	MLP dim	Params (M)	MACs (G)
ASF-former-S	2	64	64	14	384	1152	**19.3**	**5.5**
ASF-former-B	2	64	64	24	512	1536	**56.7**	**12.9**

Model	Depth N_1,N_2,N_3,N_4	Dim C	Params (M)	MACs (G)
ASF-former$_p$-S	3,4,12,5	64	**21.3**	**5.4**
ASF-former$_p$-B	3,4,18,5	96	**58.9**	**12.1**

we also design two variants correspondingly, named ASF-former$_p$-S and ASF-former$_p$-B. Their details are shown in Table II.

Training & Inference. We fix the training/inference recipe as [1] for a fair comparison. In the training phase, images are randomly cropped into size 224×224 before going through the network. We also adopt data-augmentations, such as MixUp [62], CutMix [63], Rand-Augment [64], Random-Erasing [65] to reduce over-fitting. The Exponential Moving Average (EMA) strategy is further used for training stability. We train 310 epochs using AdamW optimization, with a batch size of 512. The learning rate is initialized with 5e-4 and decreases with the cosine learning schedule. In the inference phase, images are first resized to let the short side be 256 and then center-cropped into 224×224 before being fed into the network.

C. Ablation study

In this part, we study the effectiveness of our proposed convolutional branch HMCB, Split Parallelism, Adaptive Fusion, etc. We test them on top of the *small* ASF-former-S for quick verification.

HMCB *vs* Convolutional Candidates. We plug the PCM, Residual Bottleneck, and HMCB into the ASF-former. To exclude the influence of the fusion strategy, we employ "Simple Fusion" in all three ASF-formers. The comparison is shown in Table III.

Conv Branch	Regime	Params (M)	MACs (G)	Top-1 (%)
PCM	ViTAE [14]	23.6	5.6	82.0
PCM	ASF-former	32.1	9.3	82.2
Residual Bottleneck	ASF-former	**18.3**	**5.4**	81.7
Our HMCB	ASF-former	18.8	5.5	**82.5**

We observe that the HMCB achieves the best accuracy (82.5%) among all candidates while consuming comparable

or fewer computations (Params / MACs) than the Residual Bottleneck or PCM. This validates that the HMCB is more complementary to global attention than the rest at a low computational cost. Moreover, plugging PCM into ASF-former (with simple fusion) performs slightly better than in the original ViTAE, verifying the effectiveness of the Split Parallelism mechanism. We fix the convolutional branch to be an HMCB in the following experiments.

Split Parallelism *vs* Single Path. We further compare the Split Parallelism with Single Path methods. We remove the channel split for the Single Path method and feed the entire input into an Attention-Only or HMCB-Only path. Hereby, we still adopt "Simple Fusion" (Fig. (3a)) in this ablation . Notably, the HMCB-only replaces the [CLASS] token with an average pooled vector to predict final categories.

Branch	Params (M)	MACs (G)	Top-1 (%)
Attention-only	21.5	6.1	81.7
HMCB-only	22.7	**5.2**	72.4
Attention + HMCB	**18.8**	5.5	**82.5**

The results are shown in Table IV. Our Split Parallelism achieves 82.5% accuracy, which remarkably outperforms single-path settings (81.7% for Atten-only and 72.4% for Conv-only). Thanks to the Split strategy, our parallelism achieves comparable or fewer Parameters & MACs than single path methods. This also indicates that the HMCB and attention branches are complementary; meanwhile, our Split Parallelism could capture and integrate the information from both branches very well.

Adaptive Fusion *vs* Counterparts. We implement fusion strategies in Section III-C, including "Simple Fusion", "Context-Agnostic Fusion" and "Adaptive Fusion", on top of the ASF-former. All fusion variants intake outputs from the attention branch and HMCB. We present their comparison in Table V.

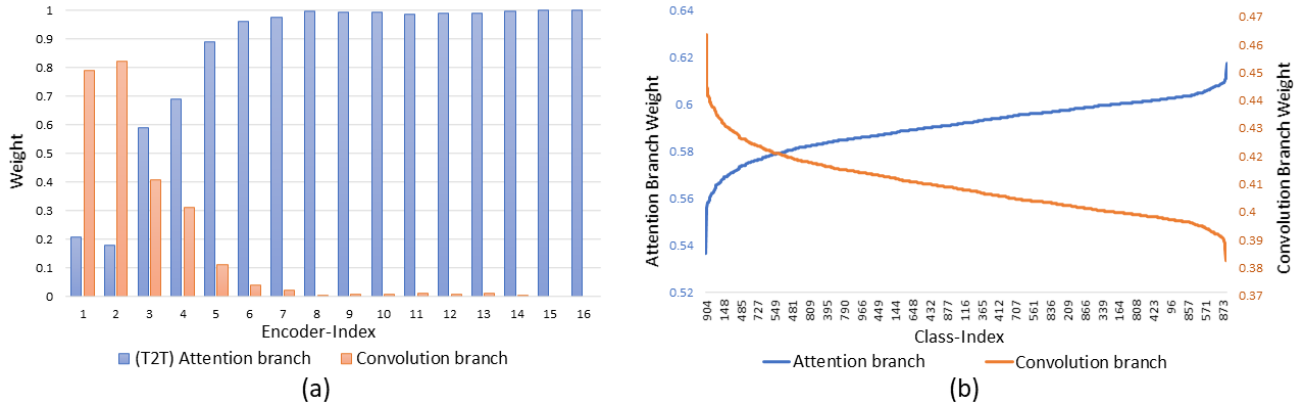| Fusion Method	Params	MACs	Top-1
Simple Fusion	18.8	5.5	82.5
Context-Agnostic Fusion	18.8	5.5	82.2
Adaptive Fusion	19.3	5.5	**82.7**

Fig. 5. The distribution of weights for HMCB and Attention branch. (a) Weights to the depth of encoder. (b) Weights to categories. (Blue/Orange denotes weights for the Attention/HMCB, this figure is best viewed in color)

We find that our Adaptive Fusion achieves 82.7% accuracy, which is superior to all the other counterparts under similar parameters and MACs. This indicates the effectiveness of adapting the weights according to visual contents and verifies the different branch preferences of different images. Notably, Context-Agnostic Fusion performs worse than Simple Fusion, showing that the coarsely learning context-agnostic weights would even degrade both branches' capability and training effect.

Effectiveness of Shortcut. We validate the influence of the shortcut (blue line in Figure (3c)) by removing it from Adaptive Fusion. The comparison is shown in Table VI.

TABLE VI
EFFECTIVENESS OF SHORTCUT ON IMAGENET VAL.

Fusion Method	Params (M)	MACs (G)	Top-1 (%)
ASF-Former-S	19.3	5.5	82.7
− shortcut	19.3	5.5	82.0 (↓ 0.7)

When discarding the skip connection, we can see that the final accuracy degrades by a large margin (0.7%) and is even much worse than Simple Fusion in Table V. This demonstrates the necessity of skip connection when fusing the outputs of two branches and verifies its ability to help the model's training by promoting gradient propagation.

Effectiveness of Pyramid Structure. We train and validate the pyramid version of ASF-former-S &ASF-former-B, named ASF-former$_p$-S & ASF-former$_p$-B, under the same recipe. The results are present in Table VII.

The underlined ASF-former$_p$-S achieves an excellent accuracy of 83.0%, which outperforms ASF-former-S by 0.3% with comparable parameters and MACs. This strongly proves the effectiveness of a multi-level pyramid structure for better feature encoding when designing hybrid transformer models. ASF-former$_p$-B achieves comparable results with ASF-former-B while consuming fewer MACs. This further indicates the computation-efficient property of our pyramid model.

TABLE VII
COMPARISON WITH DIFFERENT METHODS ON IMAGENET VALIDATION
SET.

Type	Model	Image Size	Params (M)	MACs (G)	ImageNet Top-1	Top-5
CNN (Small)	ResNet-50 [19]	224	25.6	7.6	76.7	93.3
	RegNetY-4G [66]	224	21.0	4.0	80.0	-
	ConvNeXt-T [67]	224	29.0	4.5	82.1	-
Transformer (Small)	PVT-S [58]	224	24.5	7.6	79.8	-
	DeiT-S [56]	224	22.0	4.6	79.9	95.0
	Swin-T [59]	224	28.0	4.5	81.2	95.5
	TNT-S [57]	224	23.8	5.2	81.5	95.7
	T2T-ViT-14 [1]	224	21.5	5.2	81.5	95.7
	T2T-ViT$_t$-14 [1]	224	21.5	6.1	81.7	-
Hybrid (Small)	ConT-M [7]	224	19.2	3.1	80.2	-
	ConT-B [7]	224	39.6	6.4	81.8	-
	Conformer-Ti [13]	224	23.5	5.2	81.3	-
	Swin-ACmix-T [12]	224	30.0	4.6	81.9	-
	CeiT-S [9]	224	24.2	4.5	82.0	95.9
	ViTAE-S [14]	224	23.6	5.6	82.0	95.9
	DAT-T [68]	224	29.0	4.6	82.0	-
	Focal-T [8]	224	29.1	4.9	82.2	-
	CvT-13 [6]	224	20.0	4.5	81.6	-
	CvT-21 [6]	224	32.0	7.1	82.5	-
	ASF-former-S	224	19.3	5.5	**82.7**	**96.1**
	ASF-former$_p$-S	224	21.3	5.4	**83.0**	**96.3**
CNN (Big)	ResNet-101 [19]	224	44.5	15.2	78.3	94.1
	ResNet-152 [19]	224	60.2	22.6	78.9	94.4
	RegNetY-16G [66]	224	84.0	16.0	82.9	-
	ConvNeXt-B [67]	224	89.0	15.4	83.8	-
Transformer (Big)	ViT-B/16 [2]	384	86.5	55.4	77.9	-
	ViT-L/16 [2]	384	304.3	65.8	76.5	-
	PVT-L [58]	224	61.4	19.6	81.7	-
	DeiT-B [56]	224	86.6	34.6	81.8	95.6
	T2T-ViT-24 [1]	224	64.1	14.1	82.3	-
	T2T-ViT$_t$-24 [1]	224	64.1	15.0	82.6	-
	TNT-B [57]	224	65.6	14.1	82.9	96.3
	Swin-B [59]	224	88.0	15.4	83.5	96.5
Hybrid (Big)	Conformer-S [13]	224	37.7	10.6	83.4	-
	Swin-ACmix-S [12]	224	51.0	9.0	83.5	-
	Focal-B [8]	224	89.8	16.0	83.8	-
	ASF-former-B	224	56.7	12.9	**83.9**	**96.6**
	ASF-former$_p$-B	224	58.9	12.1	**83.9**	**96.5**

D. Comparison with the state-of-the-art

We further compare the ASF-former-S and ASF-former-B with SOTAs of pure CNN, transformer, and CNN-transformer hybrid. We separately present models into *Small* and *Big* parts, considering their computation scales (Params/MACs). We only analyzed the results of ASF-former below for brevity. However, it has to be mentioned that ASF-former$_p$ performs comparably with or even better than ASF-former and further expands the advantages of our design, as shown in Table VII.

Compared with CNN SOTAs, our ASF-former outperforms the strong ConvNeXt regime in terms of accuracy with fewer parameters and comparable computations. For example, the ASF-former-S is better than the ConvNeXt-T (82.7% *vs* 82.1%) in accuracy, with much fewer parameters (19.3M *vs* 29.0M) and slightly more computations (5.5G *vs* 4.5G). And, the ASF-former-B surpasses the ConvNeXt-B (83.9% *vs* 83.8%) with much less parameters (56.7M *vs* 89.0M) and MACs (12.9G *vs* 15.4G).

Compared with pure transformer SOTAs, the ASF-former performs better than the Swin Transformer and T2T-ViT regimes in terms of accuracy, Params, with comparable MACs. Specifically, the ASF-former-S achieves higher accuracy than the Swin-T and T2T-ViT$_t$-14 (82.7% *vs* 81.2% *vs* 81.7%), with fewer parameters (19.3M *vs* 28.0M *vs* 21.5M) and comparable MACs (5.5M *vs* 4.5M *vs* 6.1M). Besides, the ASF-former-B outperforms the Swin-B and T2T-ViT$_t$-24 under all metrics: Accuracy (83.9% *vs* 83.5% *vs* 82.6%), Params (56.7M *vs* 88M *vs* 64.1M), MACs (12.9G *vs* 15.4G *vs* 15.0G).

Compared with the CNN-transformer hybrid SOTAs, the ASF-former outperforms those cascade hybrids, such as CvT and Focal Transformer, in terms of accuracy and Params, becoming the first parallel hybrid to beat cascade counterparts. Specifically, at a similar MAC scale, the ASF-former-S shows a better accuracy (82.7% *vs* 82.5% *vs* 82.2%) and fewer params (19.3M *vs* 32.0M *vs* 29.1M) than CvT-21 and Focal-T. Meanwhile, the ASF-former-B is better than Focal-B under all metrics: Accuracy (83.9% *vs* 83.8%), Params (56.7M *vs* 89.8M), MACs (12.9G *vs* 16.0G).

Among hybrid transformers, ViTAE, Swin-ACmix, Conformer, and our ASF-former all adopt parallel structure, while the ASF-former demonstrates the best accuracy at a similar computation scale. For example, the ASF-former-S and ASF-former-B separately surpass those best available, i.e., ViTAE-S and Swin-ACmix-S, by an accuracy margin of 82.7%-82.0%=0.7% and 83.9%-83.5%=0.4%. This indicates that our split parallelism with HMCB, cooperating with the adaptive fusion, can efficiently enforce the model to be lightweight and effectively boost performance via integrating parallel features.

Compared with SOTAs under ImageNet-22k pre-training, we also pre-train the model on the large-scale ImageNet-22k dataset and then fine-tune it on ImageNet-1k to further explore the upper bound of ASF-former. The classification results are shown in Table VIII. Our ASF-former achieves the same Top-1 accuracy (85.2%) as the strong SOTA of Swin-B but requires much less parameters (56.7M *vs* 88.0M) and MACs (12.9G *vs* 15.4G). These indicate the great

TABLE VIII
COMPARISON WITH DIFFERENT METHODS ON IMAGENET-1K VALIDATION
SET AFTER IMAGENET-22K PRE-TRAINING.

Model	Image Size	Params (M)	MACs (G)	ImageNet Top-1
R-101x3 [69]	384	388	204.6	84.4
ViT-B/16 [2]	384	86	55.4	84.0
ViT-L/16 [2]	384	307	190.7	**85.2**
Swin-B [59]	224	88	15.4	**85.2**
ASF-Former-S (ours)	224	**56.7**	**12.9**	85.2

TABLE IX
TRANSFERABILITY TO CIFAR-10/100.

Model	Params	CIFAR-10	CIFAR-100
ViT-B/16 [2]	86.5	98.1	87.1
ViT-L/16 [2]	304.3	97.9	86.4
T2T-ViT-14 [1]	21.5	97.5	88.4
TNT-S$_{\uparrow 384}$ [57]	23.8	98.7	90.1
ViTAE-S$_{\uparrow 384}$ [14]	23.6	98.8	90.8
CeiT-S$_{\uparrow 384}$ [9]	24.2	**99.1**	90.8
DeiT-B$_{\uparrow 384}$ [56]	86.6	**99.1**	90.8
ASF-former-S	19.3	98.7	90.4
ASF-former-B	56.7	98.8	**91.0**

capacity of the ASF-former as well as its superior computation efficiency.

E. Distribution of Weights

We plot the distribution of weights for HMCB and Attention branch with respect to the depth of encoder and categories in Fig. 5(a) and (b). For simplicity, we calculate distributions using ASF-former-S.

Figure (5a) shows the trend of weights changing with the depth of the encoder. Specifically, the ASF-former-S contains 16 encoders. For each encoder, we calculate the mean weight of the HMCB/Attention way on ImageNet-1K Val. We observe that the domination of HMCB in early encoders gradually changes to the attention when depth becomes deeper. This finding is consistent with prior works [12] that shallow layers focus on locality, whereas deep layers prefer globality, which will inspire future model designing.

Figure (5b) shows distribution weights on 1000 categories. We pick the third encoders as they are the most balanced for the HMCB/Attention (3rd depth in Fig. (5a)). We sort categories according to the descending (or increment) of HMCB (Attention) weight. We observe that categories prefer locality/globality differently, which is also affected by the depth of the encoder. Specifically, small objects like "*Ruler*" and "*Can opener*" prefer the locality. Categories like "*Honeycomb*" also produce large weights for the convolution branch for its requirement of capturing the local texture & edge information. Sceneries or large objects like "*Triumphal arch*", "*Kimono*" and "*Leonberg*" prefer global information.

F. Transferability to downstream datasets

To investigate the transferability of our ASF-former, we further fine-tune the proposed models on CIFAR-10 and CIFAR-

TABLE X
TRANSFERABILITY TO OBJECT DETECTION AND INSTANCE
SEGMENTATION TASK.

Backbone	Method	Lr Schd	Params (M)	Box mAP	Mask mAP
ResNet-50 [19]	Mask RCNN	1x	44	38.2	34.7
PVT-S [58]	Mask RCNN	1x	44	40.4	37.8
ResT-Base [70]	Mask RCNN	1x	50	41.6	38.7
Twins-PCSVT-S [71]	Mask RCNN	1x	44	42.9	40.0
Twins-PVT-S [71]	Mask RCNN	1x	44	43.4	40.3
RegionViT-S+ [72]	Mask RCNN	1x	51	43.5	40.4
Conformer-S [13]	Mask RCNN	1x	58	43.6	39.7
Swin-T [59]	Mask RCNN	1x	48	43.7	39.8
DAT-T [68]	Mask RCNN	1x	48	44.4	40.4
ViTAE-S [14]	Mask RCNN	1x	**37**	44.6	40.2
Focal-T [8]	Mask RCNN	1x	49	44.8	41.0
ASF-Former$_p$-S (ours)	Mask RCNN	1x	38	**45.9**	**41.2**
ResNet-50 [19]	Cascade RCNN	1x	82	41.2	35.9
Swin-T [59]	Cascade RCNN	1x	86	48.1	41.7
ViTAE-S [14]	Cascade RCNN	1x	**75**	48.9	42.0
DAT-T [68]	Cascade RCNN	1x	86	49.1	42.5
ASF-Former$_p$-S (ours)	Cascade RCNN	1x	79	**49.7**	**42.8**

TABLE XI
TRANSFERABILITY TO SEMANTIC SEGMENTATION TASK.

Backbone	Method	Lr Schd	Params (M)	mIoU	mIoU (MS)
ResNet-50 [19]	UPerNet	160K	67	42.1	42.9
Swin-T [59]	UPerNet	160K	60	44.5	45.8
Swin-ACmix-T [12]	UPerNet	160K	60	45.3	-
DAT-T [68]	UPerNet	160K	60	45.5	46.4
ViTAE-S [14]	UPerNet	160K	**49**	45.4	**47.8**
DW-T [74]	UPerNet	160K	61	45.7	46.9
Focal-T [8]	UPerNet	160K	62	45.8	47.0
ConvNeXt-T [67]	UPerNet	160K	60	46.0	46.7
Twins-SVT-S [71]	UPerNet	160K	54	46.2	47.1
Twins-PCPVT-S [71]	UPerNet	160K	55	46.2	47.5
ASF-Former$_p$-S (ours)	UPerNet	160K	51	**46.7**	47.6

100 datasets. The initial learning rate is 0.025 for CIFAR-10 and 0.05 for CIFAR-100.

The validation results are shown in Table IX. Our ASF-former achieves comparable results on CIFAR-10 and the state-of-the-art results on CIFAR-100 under 224×224 resolution, showing its superior transferability.

G. Transferability to object detection & instance segmentation

As described in Section III-D, ASF-former$_p$ is adjusted from the ASF-former, to make the transformer more compatible with the detection & segmentation tasks. We test the performance of our ASF-former$_p$ on standard object detection and instance segmentation benchmarks: MS-COCO 2017 [73] datasets. Specifically, the COCO contains 118k training images and 5k validation images. During experiments, we adopt Mask RCNN and Cascade RCNN as the detection frameworks and follow the same experimental settings as Swin [59] for a fair comparison. Specifically, multi-scale training and AdamW optimizer are used. The initial learning rate is set to be 0.0001 with a weight decay of 0.05. We use the batch size of 16 and 1x schedule (12 epochs).

The experimental results are shown in Table X. We compare our ASF-former$_p$ with different kinds of vision backbones, including the most widely-used pure CNN, transformer, and the recent strong hybrid models. Among them, ASF-former$_p$ achieves the best results on both object detection (Box mAP) and instance segmentation (Mask mAP) tasks under two widely-used detection frameworks with comparable or fewer parameters. It indicates the superior capacity of ASF-former$_p$ as a backbone for downstream tasks.

H. Transferability to semantic segmentation

We further validate our ASF-former$_p$ as the backbone model for semantic segmentation tasks using the standard ADE20K [75] dataset. It contains more than 20k images in the training set and 2k images in the validation set. Following

Swin, we take the widely-used UPerNet as the framework and use the AdamW optimizer for training. The initial learning rate is set to be 0.00006 with a weight decay of 0.01, and the batch size is 16. We follow the 160K lr schedule (160k iterations) for a fair comparison.

The results are shown in Table XI which include the standard and multi-scale (MS) testing results. We can see that ASF-former$_p$ achieves a mIoU of 46.7 in standard testing, which outperforms the other backbone models by a large margin with comparable or fewer parameters. Under a multi-scale augmentation setting, ASF-former$_p$ gets comparable results to the SOTA methods. These show that our method works well as the backbone in segmentation tasks and proves its excellent transferability to downstream vision tasks.

V. CONCLUSION

In this paper, we propose a novel hybrid transformer called ASF-former. It adopts Split Parallelism, which splits channels into half for two-path inputs. It introduces the novel HMCB that complements the attention and the Adaptive Fusion for feature merging. We experimentally verified that the three mechanisms could a good balance of efficacy and efficiency and achieve SOTA results. We also validate that the role of local/global information varies with respect to visual categories and network depth. Besides, this hybrid design also achieves promising results on downstream tasks, including object detection and segmentation.

REFERENCES

[1] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. H. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 2021, pp. 538–547.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[3] Z. Chen, L. Xie, J. Niu, X. Liu, L. Wei, and Q. Tian, "Visformer: The vision-friendly transformer," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 2021, pp. 569–578.

[4] A. Hassani, S. Walton, N. Shah, A. Abuduweili, J. Li, and H. Shi, "Escaping the big data paradigm with compact transformers," *CoRR*, vol. abs/2104.05704, 2021.

[5] S. Mehta and M. Rastegari, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," *CoRR*, vol. abs/2110.02178, 2021.

[6] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 2021, pp. 22–31.

[7] H. Yan, Z. Li, W. Li, C. Wang, M. Wu, and C. Zhang, "Contnet: Why not use convolution and transformer at the same time?" *CoRR*, vol. abs/2104.13497, 2021.

[8] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal self-attention for local-global interactions in vision transformers," *CoRR*, vol. abs/2107.00641, 2021.

[9] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu, "Incorporating convolution designs into visual transformers," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 2021, pp. 559–568.

[10] H. He, J. Liu, Z. Pan, J. Cai, J. Zhang, D. Tao, and B. Zhuang, "Pruning self-attentions into convolutional layers in single path," *CoRR*, vol. abs/2111.11802, 2021.

[11] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, and Z. Liu, "Mobile-former: Bridging mobilenet and transformer," *CoRR*, vol. abs/2108.05895, 2021.

[12] X. Pan, C. Ge, R. Lu, S. Song, G. Chen, Z. Huang, and G. Huang, "On the integration of self-attention and convolution," *CoRR*, vol. abs/2111.14556, 2021.

[13] Z. Peng, W. Huang, S. Gu, L. Xie, Y. Wang, J. Jiao, and Q. Ye, "Conformer: Local features coupling global representations for visual recognition," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 2021, pp. 357–366.

[14] Y. Xu, Q. Zhang, J. Zhang, and D. Tao, "Vitae: Vision transformer advanced by exploring intrinsic inductive bias," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021, pp. 28 522–28 535.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, 2012, pp. 1106–1114.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 1–9.

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 2818–2826.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 770–778.

[20] ——, "Identity mappings in deep residual networks," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, vol. 9908, 2016, pp. 630–645.

[21] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 2261–2269.

[22] R. B. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015, pp. 1440–1448.

[23] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015, pp. 91–99.

[24] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 779–788.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, vol. 9905, 2016, pp. 21–37.

[26] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 2999–3007.

[27] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 936–944.

[28] Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 6154–6162.

[29] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 2980–2988.

[30] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017.

[31] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018.

[32] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 1933–1941.

[33] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 5534–5542.

[34] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 4724–4733.

[35] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*, vol. 11205, 2018, pp. 831–846.

[36] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 7794–7803.

[37] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 2019, pp. 6201–6210.

[38] J. Lin, C. Gan, and S. Han, "TSM: temporal shift module for efficient video understanding," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 2019, pp. 7082–7092.

[39] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.

[40] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 4510–4520.

[41] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 6848–6856.

[42] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, vol. 97, 2019, pp. 6105–6114.

[43] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 7132–7141.

[44] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," in *Advances*

in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018, pp. 9423–9433.

[45] Y. Hao, H. Zhang, C.-W. Ngo, and X. He, "Group contextualization for video recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 928–938.

[46] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019, pp. 510–519.

[47] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. J. Smola, "Resnest: Split-attention networks," *CoRR*, vol. abs/2004.08955, 2020.

[48] S. Alfasly, C. K. Chui, Q. Jiang, J. Lu, and C. Xu, "An effective video transformer with synchronized spatiotemporal and spatial self-attention for action recognition," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[49] X. Lin, S. Sun, W. Huang, B. Sheng, P. Li, and D. D. Feng, "Eapt: Efficient attention pyramid transformer for image processing," *IEEE Transactions on Multimedia*, pp. 1–1, 2021.

[50] W. Tang, F. He, and Y. Liu, "Ydtr: Infrared and visible image fusion via y-shape dynamic transformer," *IEEE Transactions on Multimedia*, pp. 1–16, 2022.

[51] M. Li, J. Liu, C. Zheng, X. Huang, and Z. Zhang, "Exploiting multi-view part-wise correlation via an efficient transformer for vehicle re-identification," *IEEE Transactions on Multimedia*, pp. 1–1, 2021.

[52] J. Zhao, H. Wang, Y. Zhou, R. Yao, S. Chen, and A. El Saddik, "Spatial-channel enhanced transformer for visible-infrared person re-identification," *IEEE Transactions on Multimedia*, pp. 1–1, 2022.

[53] Y. Zhou, Y. Wang, and L.-P. Chau, "Moving towards centers: Re-ranking with attention and memory for re-identification," *IEEE Transactions on Multimedia*, 2022.

[54] J. Wang, S. Chen, Z. Wu, and Y.-G. Jiang, "Ft-tdr: Frequency-guided transformer and top-down refinement network for blind face inpainting," *IEEE Transactions on Multimedia*, pp. 1–1, 2022.

[55] W. Li, H. Liu, R. Ding, M. Liu, P. Wang, and W. Yang, "Exploiting temporal contexts with strided transformer for 3d human pose estimation," *IEEE Transactions on Multimedia*, pp. 1–1, 2022.

[56] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, vol. 139, 2021, pp. 10347–10357.

[57] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021, pp. 15908–15919.

[58] W. Wang, E. Xie, X. Li, D. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 2021, pp. 548–558.

[59] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, 2021, pp. 9992–10002.

[60] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 2009, pp. 248–255.

[61] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[62] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[63] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 2019, pp. 6022–6031.

[64] E. D. Cubuk, B. Zoph, J. Shlens, and Q. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[65] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, 2020, pp. 13001–13008.

[66] I. Radosavovic, R. P. Kosaraju, R. B. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020, pp. 10425–10433.

[67] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[68] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 4794–4803.

[69] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, vol. 12350, 2020, pp. 491–507.

[70] Q. Zhang and Y. Yang, "Rest: An efficient transformer for visual recognition," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021, pp. 15475–15485.

[71] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen, "Twins: Revisiting the design of spatial attention in vision transformers," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021, pp. 9355–9366.

[72] C. Chen, R. Panda, and Q. Fan, "Regionvit: Regional-to-local attention for vision transformers," *CoRR*, vol. abs/2106.02689, 2021.

[73] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, vol. 8693, 2014, pp. 740–755.

[74] P. Ren, C. Li, G. Wang, Y. Xiao, Q. Du, X. Liang, and X. Chang, "Beyond fixation: Dynamic window visual transformer," *CoRR*, vol. abs/2203.12856, 2022.

[75] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, 2019.