Adaptive Split-Fusion Transformer

Zixuan Su$^{1,2}$, Jingjing Chen$^{1,21}$, Lei Pang$^2$, Chong-Wah Ngo$^4$, Yu-Gang Jiang$^{1,2}$

$^1$Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University
$^2$Shanghai Collaborative Innovation Center of Intelligent Visual Computing
$^3$City University of Hong Kong $^4$Singapore Management University
xzsu21@m.fudan.edu.cn, {chenjingjing, ygj}@fudan.edu.cn, cactuslei@gmail.com, cwngo@smu.edu.sg

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 two parts: a convolution branch and an attention branch. With these, 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 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 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 carefully craft an efficient convolution path named Half-Residual Mobile Convolutional Branch (HMCB) which 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. With these, the Split Parallelism shares a similar complexity as single-path (convolution or attention) models.

Adaptive Fusion module intakes visual feature 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, which complements the attention feature with high cost-effectiveness.

- **Adaptive Fusion.** We 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 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

**Vision Transformer.** Transformer receives extensive interest in vision 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 classification accuracy, researchers spare more effort on developing new transformers, including DeiT [15], T2T-ViT [1], PVT [16], Swin [17] than before. Specifically, DeiT [15] adopted convnet as a teacher and trained transformer under a teacher-student strategy, thus lowering the requirement for large-scale training data. It relies on the distillation token to introduce locality into a transformer, thus lowering the requirement for large-scale training data. T2T-ViT [1] designed the T2T module to shrink token length via concatenating features of local neighboring pixels. TNT [28] altered the granularity of patch dividing by further splitting local patches into sub-patches. The inner attention will be calculated within each patch before outer global attention. In order to be ported to various downstream tasks, PVT [16] introduced a progressive shrinking pyramid structure and a spatial-reduction attention for learning multi-scale and high-resolution features. For further parameter and computation efficiency, Swin Transformer [17] 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 Fig. (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:

\[
N_0 = \left\lfloor \frac{H + 2p - k}{k - o} + 1 \right\rfloor \times \left\lfloor \frac{W + 2p - k}{k - o} + 1 \right\rfloor .
\]

Tokens \( T_0 \) go through the reduction and computation stages for representation learning.

Reduction stage contains \( L_1 \) replicated ASF-R + T2T pairs, where they 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 \( \hat{T}_i \in \mathbb{R}^{N_i \times D'} \). The token-length \( N_i \) and dimension \( D_i \) would reduce and increase according to the depth \( i \in [1, 2, \ldots, L_1] \), 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:

\[
\hat{T}_{i-1} = f_{ar}(T_{i-1})
\]
\[
T_i = f_{tl2}(\hat{T}_{i-1})
\]

where \( f_{ar}(\cdot) \) and \( f_{tl2}(\cdot) \) 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.

\[
T_{out} = \text{Linear}(T_{L_1})
\]

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.

\[
X_0 = [T_{out}; C_0]
\]

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

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

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

\[
Y = \text{Linear}(C_{L_2}), \quad Y \in \mathbb{R}^{Categories}
\]

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 sub-tensor \( T^{(a)}/T^{(b)} \) 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:

\[
\hat{T}^{(a)} = \text{img2seq}(f_{convb}(\text{seq2img}(T^{(a)})))
\]
\[
\hat{T}^{(b)} = f_{attach}(T^{(b)})
\]

where \( f_{attach}(\cdot) \) 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 craft an efficient convolutional branch named Half-Residual Mobile Convolutional Branch and present it in Section III-B.

Adaptive Fusion (Fig (1) bottom right) performs weighted sum on tensors processed by the two paths with adaptive scalars \( \alpha \) and \( \beta \). The scalars are calculated according to visual context by Eq. (11)\textasciitilde(12), inspired by SKNet [18].

\[
S = \hat{T}^{(a)} + \hat{T}^{(b)}
\]
\[
\alpha = \text{Sigmoid}(\text{Linear}^2(\text{Pool}(S)))
\]
\[
\beta = 1 - \alpha
\]

\[
\hat{T} = \alpha \cdot \hat{T}^{(a)} + \beta \cdot \hat{T}^{(b)} + S
\]

where Eq. (11) performs global average pooling (Pool) before adopts two fully-connected layers (Linear), with BatchNorm and GelU activations in between. Notably, we generate the \( \alpha \& \beta \) in a Sigmoid way. Though this way is theoretically equivalent to a Softmax function, it is practically simple in implementation. To avoid that the gradient vanishment deactivates a particular branch when \( W_\alpha \) or \( W_\beta \) falls into extremely tiny value, we add an extra skip connection (blue line) to enforce gradients to be propagated to both ways. We introduce and compare various fusion strategies in Section 1.2 of Supplementary and observe a significant improvement with our Adaptive Fusion strategy.

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 $\hat{T}/\hat{X}$ of the ASF-R/C encoders is generated as in Eq. (14)∼(15), where $f_{mlp}()$ denotes the MLP with two $fc$ layers and a GeLU activation:

$$
\hat{T} = f_{mlp}(\hat{T}) + \hat{T} \quad (14)
$$

$$
\hat{X} = f_{mlp}(\hat{X}) + \hat{X}, \quad \hat{X} = Conv\left(\hat{X}\right) + X \quad (15)
$$

### B. Half-Residual Mobile Convolutional Branch

Half-Residual Mobile Convolutional Branch (HMCB) is modified based on PCM [14], with the help of efficient conv approximations (Fig (1) bottom left). Inspired by MobileNet [19], we first factorize each conventional $3 \times 3$ conv into one $3 \times 3$ depth-wise conv followed by one $1 \times 1$ conv and then we add another $1 \times 1$ conv before the first depth-wise conv.

The HMCB is more complementary to the attention way than its counterparts while consuming fewer 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 promote the training across channels. We introduce and comprehensively compare three designs (PCM, Residual Bottleneck and HMCB) in Section 1.1 of Supplementary and observe that our HMCB performs the best under all metrics.

### IV. EXPERIMENTS

We evaluate the ASF-former on standard ImageNet-1K benchmarks, with metrics like Top-1/5 accuracy, model Params, and inferencing MACs. Experimental results validate the efficacy and efficiency of the ASF-former. In Supplementary, we further conduct experiments on downstream datasets/tasks and visualize the local/global preference.

#### A. Experimental settings

**Dataset.** We conduct ablation experiments on ImageNet-1K dataset [20]. It 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.

**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 cost, 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)).

**Training & Inference.** We fix the training/inference recipe as [1] for a fair comparison. In the training phase, images are randomly cropped into size $224 \times 224$ before going through the network. We adopt data-augmentations such as MixUp [21], CutMix [22], Rand-Augment [23], Random-Erasing [24] to reduce over-fitting. The Exponential Moving Average (EMA) 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 $224 \times 224$ before calculating.

**B. Ablation study**

In this part, we study the effectiveness of our proposed module. We test them on top of the small ASF-former-S for quick verification. More Ablation studies including concolutional branch and fusion strategies are introduced in Supplementary.

**Split Parallelism vs Single Path.** We 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 adopt “Simple Fusion” (Fig.2(a) in Supplementary) in this ablation. Notably, the HMCB-only replaces the [CLASS] token with an average-pooled vector to predict final categories.

### TABLE I

| Model            | Reduction stage | Computation stage | Model Size |
|------------------|-----------------|-------------------|------------|
|                  | $L_1$ | Token dim | MLP dim | $L_2$ | Token dim | 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     |

### TABLE II

| 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 II. 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 indicates that HMCB and attention branch are complementary and our Split Parallelism could well integrate the information from both branches.

**Effectiveness of Shortcut.** We validate the influence of the shortcut (blue line in Fig (1) bottom right)) by removing it from Adaptive Fusion. The comparison is shown in Table III.

| 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) |
explore the upper bound of ASF-former. As shown in Table V, 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 capacity of the ASF-former as well as its superior computation efficiency.

V. CONCLUSION

In this paper, we propose a novel hybrid transformer called ASF-former. It adopts Split Parallelism to 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 show 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/datasets.

REFERENCES

[1] Li Yuan, Yinpeng Chen, Tao Wang, Weihao Yu, Yujin Shi, Zihang Jiang, Francis E. H. Tay, Jiaxì Feng, and Shucheng Yan, “Tokens-to-token vits: Training vision transformers from scratch on imagenet,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 538–547.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvaın Gelly, Jakob Uszkoreit, and Neil Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in International Conference on Learning Representations, 2021.

[3] Z. Chen, L. Xie, J. Niu, X. Liu, L. Wei, and Q. Tian, “Visformer: The vision-friendly transformer,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 569–578.

[4] A. Hassani, S. Walton, N. Shah, A. Abudweiti, J. Li, and H. Shi, “Escaping the big data paradigm with compact transformers,” arXiv preprint arXiv:2104.05704, 2021, unpublished.

[5] S. Mehta and M. Rastegari, “Mobivit: Light-weight, general-purpose, and mobile-friendly vision transformer,” in International Conference on Learning Representations, 2021.

[6] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang, “Cvt: Introducing convolutions to vision transformers,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 22–31.

[7] Haotian Yan, Zhe Li, Weijian Li, Changhu Wang, Ming Wu, and Chuang Zhang, “Convitnet: Why not use convolution and transformer at the same time?” arXiv preprint arXiv:2104.13497, 2021, unpublished.

[8] Jianwei Yang, Chunjuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao, “Focal self-attention for local-global interactions in vision transformers,” arXiv preprint arXiv:2107.00641, 2021, unpublished.

[9] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Yu, “Incorporating convolution designs into visual transformers,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 559–568.

[10] Haoyu He, Jing Liu, Zixheng Pan, Jianfei Cai, Jing Zhang, Dacheng Tao, and Bohan Zhuang, “Pruning self-attentions into convolutional layers in single path,” arXiv preprint arXiv:2111.11802, 2021, unpublished.

[11] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, and Z. Liu, “Mobile-former: Bridging mobilenet and transformer,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5260–5269.

[12] Xuran Pan, Chunjiang Ge, Rui Lu, Shiji Song, Guanfu Chen, Zeyi Huang, and Gao Huang, “On the integration of self-attention and convolution,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 805–815.