FasterViT: Fast Vision Transformers with Hierarchical Attention

Ali Hatamizadeh, Greg Heinrich, Hongxu Yin, Andrew Tao, Jose M. Alvarez,
Jan Kautz, Pavlo Molchanov
NVIDIA
{ahatamizadeh, gheinrich, dannyy, atao, josea, jkautz, pmolchanov}@nvidia.com

Abstract

We design a new family of hybrid CNN-ViT neural networks, named FasterViT, with a focus on high image throughput for computer vision (CV) applications. FasterViT combines the benefits of fast local representation learning in CNNs and global modeling properties in ViT. Our newly introduced Hierarchical Attention (HAT) approach decomposes global self-attention with quadratic complexity into a multi-level attention with reduced computational costs. We benefit from efficient window-based self-attention. Each window has access to dedicated carrier tokens that participate in local and global representation learning. At a high level, global self-attentions enable the efficient cross-window communication at lower costs. FasterViT achieves a SOTA Pareto-front in terms of accuracy vs. *image throughput. We have extensively validated its effectiveness on various CV tasks including classification, object detection and segmentation. We also show that HAT can be used as a plug-and-play module for existing networks and enhance them. We further demonstrate significantly faster and more accurate performance than competitive counterparts for images with high resolution. Code is available at* https://github.com/NVlabs/FasterViT.

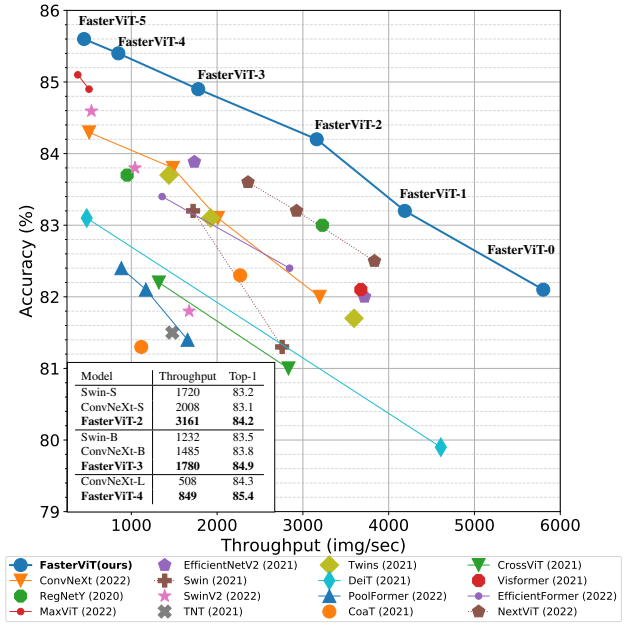


Figure 1. Comparison of image throughput and ImageNet-1K Top-1 accuracy. The family of FasterViT models achieves a new Pareto front encompassing various throughput and Top-1 accuracy trade-offs. Specifically, FasterViT has significantly better throughput compared to other ViT-based models. For all models, throughput is measured on A100 GPU with batch size of 128.

1. Introduction

Vision Transformers (ViTs) [18] have recently become popular in computer vision and achieved superior performance in various applications such as image classification [38, 17, 35], object detection [77, 21] and semantic segmentation [61, 10]. In addition to learning more uniform local and global representations across their architecture when compared to Convolutional Neural Networks (CNNs), ViTs scale properly to large-scale data and model sizes [47, 45]. Recently, several efforts [27, 63] have also shown the exceptional capability of ViTs in self-supervised learning of surrogate tasks such as masked image modeling which may significantly enhance the performance of downstream applications. Despite these advantages, lack of inductive bias in pure ViT models may require more training data and impede performance [65]. Hybrid architectures, which consist of both CNN and ViT-based components, could address this problem and achieve competitive performance without needing large-scale training datasets [18] or other techniques such as knowledge distillation [52].

An integral component of ViTs is the self-attention mechanism [56, 18] which enables modeling of both short and long-range spatial dependencies. However, the quadratic computational complexity of self-attention significantly impacts the efficiency and hinders its use for applications with high-resolution images. In addition, contrary to the isotropic architecture (*i.e.*, same feature resolution with no downsam-

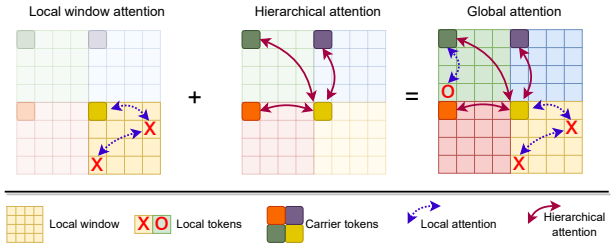


Figure 2. Visualization of the proposed Hierarchical Attention in the feature space. By performing local window attention and hierarchical attention we can achieve global information propagation at reduced costs.Best viewed in color.

pling) of the original ViT model, learning feature representations in a multi-scale manner typically yields better performance [20, 58], specifically for downstream applications (*e.g.*, detection, segmentation).

To address these issues, Swin Transformer [38] proposed a multi-scale architecture in which self-attention is computed in local windows, and window-shifting allows for interaction of different regions. However, due to the limited receptive field of these local regions and small area of coverage in window shifting [38, 35], capturing cross-window interactions and modeling the long-range spatial dependencies become challenging for large-resolution input features. Furthermore, using self-attention blocks in early stages with larger resolution may impact the image throughput due to the increased number of local windows. Recently, the Swin Transformer V2 model [37] was proposed to address training instabilities on high-resolution images by improving the self-attention mechanism. However, in addition to having a lower image throughput compared to the Swin Transformer [38], Swin Transformer V2 still relies on the original window-shifting mechanism for cross-interaction of different windows, which becomes less effective with large image sizes.

In this work, we attempt to address these issues and propose a novel hybrid architecture, denoted as FasterViT, which is tailored for high-resolution input images, while maintaining a fast image throughput. FasterViT consists of four different stages in which the input image resolution is reduced by using a strided convolutional layer, while doubling the number of feature maps. We propose to leverage residual convolutional blocks in the high-resolution stages of the architecture (*i.e.*, stage 1, 2), while employing transformer-blocks in later stages (*i.e.*, stage 3, 4). This strategy allows for fast generation of high-level tokens which can be further processed with the transformer-based blocks. For each transformer block, we use an interleaved pattern of local and, newly proposed, Hierarchical Attention blocks to capture both short and long-range spatial dependencies and efficiently model the cross-window interactions. Specifically, our proposed Hierarchical Attention (see Fig. 2) learns carrier tokens as a summary of each local window and efficiently models the cross-interaction between these regions. The computational complexity of the Hierarchical Attention grows almost linearly with input image resolution, as the number of regions increases, due to the local windowed attention being the compute bottleneck. Hence, it is an efficient, yet effective way of capturing long-range information with large input features.

We have extensively validated the effectiveness of the proposed FasterViT model on various image tasks and datasets such as ImageNet-1k for image classification, MS COCO for object detection and instance segmentation and ADE20K dataset for semantic segmentation. FasterViT achieves state-of-the-art performance considering the trade-off between performance (*e.g.*, ImageNet-1K top-1 accuracy) and image throughput (see Fig. 1). To demonstrate the scalability of FasterViT for larger datasets, we have also pre-trained FasterViT on ImageNet-21K dataset and achieved state-of-the-art performance when fine-tuning and evaluating on larger-scale resolutions.

The summary of our contributions is as follows:

- We introduce FasterViT, which is a novel hybrid vision transformer architecture designed for an optimal trade-off between performance and image throughput. FasterViT scales effectively to higher resolution input images for different dataset and model sizes.
- We propose the Hierarchical Attention module which efficiently captures the cross-window interactions of local regions and models the long-range spatial dependencies.
- FasterViT achieves a new SOTA Pareto front in terms of image throughput and accuracy trade-off and is significantly faster than comparable ViT-based architectures yielding significant speed-up compared to recent SOTA models. It also achieves competitive performance on downstream tasks of detection and instance segmentation on MS COCO dataset and semantic segmentation on ADE20K dataset.

2. Related Work

Vision Transformers. Oriented from the language processing domain, the first application of transformer architecture to vision task immediately offers an inspiring demonstration of the high efficacy of attention across image patches across varying scenarios [18]. The appealing strength of vision transformer and its architecture and logic simplicity has therefore triggered a quickly evolving literature in the past two years, where ViT performance is quickly boosted by an erupting new set of innovations: network-wise leveraging knowledge distillation for data-efficient training as in DeiT [52], hybriding convolution and self-attention for enhanced inductive biases as in LeViT [24], imposing CNN-inspired pyramid rules on ViTs [57, 58], along with component-wise improvements such as improved token uti-

lization as in T2T-ViT [72], enhanced positional embedding [12], local window attention as shown in the inspiring work of the Swin family [38, 37] and CSwin [17], global attention in GCViT [26], among many other architectural insights [11, 76, 73]. Along with the increasing capacity comes the increasing computation burden. As similarly facing challenges in scaling up the models in language tasks (e.g., from BERT-Large 0.3B [16], to Megatron-LM 8.3B [50], and Switch-Transformer1.6T [22]), scaling up vision transformers is also a highly challenging but highly important task [14, 37] due to the attention-extensive nature of transformers, urging efficiency for pervasive usage.

Towards Enhanced Efficiency. Boosting up ViT efficiency has therefore been a very vibrant area. One stream of approach roots in the efficient deep learning literature that cuts down on network complexity leveraging popular methods such as efficient attention [3, 41, 4], network compression [7, 8, 34, 67], dynamic inference [69, 48], operator adaptation [43], token merging and manipulations [42, 66], etc. These methods can yield off-the-shelf speedups on target ViT backbones, but are also limited to the original backbone's accuracy and capacity. Another stream of work, on the other hand, focuses on designing new ViT architectures with enhanced efficiency as an original design objective. For example, EfficientFormer [33] entails mobile applications through dimension-consistent re-design of transformer block and removing redundant architectural components. VisFormer [9] transits computation extensive transformer to a convolutional counterpart for enhanced vision efficiency. CrossViT [5] learns multi-scale features and utilizes small/large-patch backed tokens that are channeled by efficient attention, offering linear time and memory complexity. Even with such a rapid progress in literature, enabling efficient ViTs remains a significant challenge, where we next further push the Pareto front of faster ViT on top of prior art by a large margin. Note that we focus on the second stream of architectural redesign for efficiency boost, and consider a joint exploration with the first acceleration stream of method like compression as orthogonal and fruitful future work.

Global Self-Attention. A number of efforts have introduced global self-attention to capture more contextual information. In NLP (*i.e., 1D*), BigBird [74] and LongFormer [2] proposed to select special tokens (*i.e. non-learnable*) as global tokens to attend to other tokens via a sliding-window dense self-attention. In computer vision, EdgeViT [44], Twins [11] and Focal Transformer [68] proposed hierarchical-like attention mechanisms which rely on heuristic token aggregation in the forms of pooling [68] or linear projection [44, 11]. There are three key differences between these efforts and our proposed hierarchical attention: (1) as opposed to using a pre-defined mechanism to select the global tokens (*e.g., random*), we propose to learn these tokens (*i.e., carrier token*) via summarizing the role of each region in the input feature space (2) we propose learnable token aggregation and propagation mechanisms by computing self-attention among carrier tokens (3) as opposed to using dense/dilated self-attention, our proposed HAT uses local window-based self-attention and has a smaller computational complexity.

3. FasterViT

3.1. Design Principals

We next detail our FasterViT architecture, offering Pareto accuracy-latency trade-off. We focus on highest throughput for computer vision tasks on mainstream off-the-shelf hardware such as GPUs that excel in parallel computing. Computation in this case involves a set of streaming multiprocessors (SMs) with CUDA and Tensor cores as computation units. It requires frequent data transfer for calculation and can be impacted by data movement bandwidth. As such, operations bounded by computation are math-limited, while those bounded by memory transfer are memory-limited. It requires a careful balance between the two to maximize throughput.

In hierarchical vision models, spatial dimension of intermediate representation shrinks as inference proceeds. Initial network layers mostly have larger spatial dimensions and fewer channel (*e.g.*, $112\times112\times64$), making them memory-bound. This makes a better fit for compute-intensive operations, such as dense convolution instead of depth-wise/sparse counterparts that impose extra transfer cost. Also operations not representable in matrix manipulation forms, *e.g.*, non-linearity, pooling, batch normalization, are also memory-bound and shall be minimized for usage. On the contrary, later layers tend to be math-limited with computationally expensive operations. For example, hierarchical CNNs have feature maps of size 14×14 with high dimensional kernels. This leaves room for more expressive operations such as Layer Normalization, squeeze-and-excitation, or attention, with fairly small effect on throughput. Guided by these insights we design a novel architecture that will benefit all stages from accelerated computing hardware.

3.2. Architecture

Our overall design is shown in Fig. 3. It exploits convolutional layers in the earlier stages that operate on higher resolution. The second half of the model relies on novel hierarchical attention layers to reason spatially across the entire feature maps. In this design, we optimize the architecture for compute and throughput. As a result, the first half of the network and downsampling blocks make use of dense convolutional kernels. We also avoid squeeze-and-excitation operators and minimize Layer Normalization for higher resolution stages (*i.e.*, 1, 2), as these layers tend to be math-limited. Later stages (*i.e.*, 3, 4) in the architecture tend to be math-limited as GPU hardware spends more time on compute compared to the memory transfer cost. As a result,

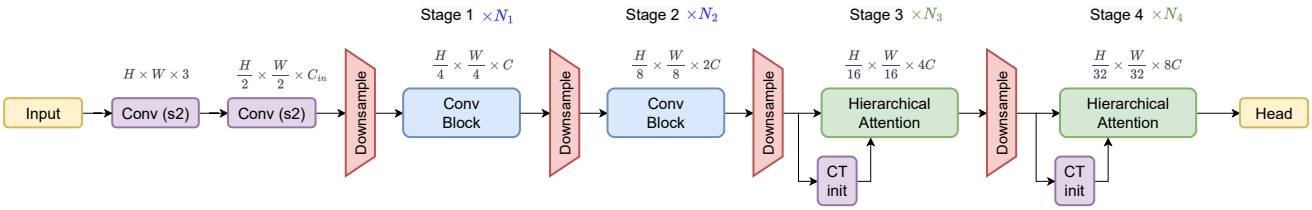


Figure 3. Overview of the FasterViT architecture. We use a multi-scale architecture with CNN and transformer-based blocks in stages 1, 2 and 3, 4, respectively. Best viewed in color.

applying multi-head attention will not be a bottleneck.

3.3. FasterViT Components

Stem An input image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ is converted into overlapping patches by two consecutive 3×3 convolutional layers, each with a stride of 2, which project them into a D-dimensional embedding. The embedded tokens are further batch-normalized [32] and use the ReLU activation function after each convolution.

Downsampler Blocks FasterViT follows the hierarchical structure: the spatial resolution is reduced by 2 between stages by a downsampling block. We apply 2D layer normalization on spatial features, followed by a convolutional layer with a kernel of 3×3 and a stride of two.

Conv Blocks Stage 1 and 2 consist of residual convolutional blocks, which are defined as

$$\begin{aligned} \hat{\mathbf{x}} &= \text{GELU}(\text{BN}(\text{Conv}_{3\times3}(\mathbf{x}))), \\ \mathbf{x} &= \text{BN}(\text{Conv}_{3\times3}(\hat{\mathbf{x}})) + \mathbf{x}, \end{aligned} \tag{1}$$

where BN denotes batch normalization [32]. Following the design principles, these convolutions are dense.

Hierarchical Attention In this work, we propose a novel formulation of windowed attention, summarized in Fig 2 and detailed presentation in Fig 4. We start with local windows introduced in Swin Transformer [38]. Then, we introduce a notion of *carrier tokens* (CTs) that play the summarizing role of the entire local window. The first attention block is applied on CTs to summarize and pass global information. Then, local window tokens and CTs are concatenated, such that every local window has access only to its own set of CTs. By performing self attention on concatenated tokens we facilitate local and global information exchange at reduced cost. By alternating sub-global (CTs) and local (windowed) self-attention we formulate a concept of *hierarchical attention*. Conceptually, CTs can be further grouped into windows and have a higher order of carrier tokens.

Assume we are given an input feature map $\mathbf{x} \in \mathbb{R}^{H \times W \times d}$ in which H, W and d denote the height, width and number of feature maps, let us set $H = W$ for simplicity. We first partition the input feature map into $n \times n$ local windows with $n = \frac{H^2}{k^2}$, where k is the window size, as:

$$\hat{\mathbf{x}}_\mathbf{l} = \text{Split}_{k\times k}(\mathbf{x}). \tag{2}$$

The key idea of our approach is the formulation of *carrier tokens* (CTs) that help to have an attention footprint much larger than a local window at low cost. At first, we initialize CTs by pooling to $L = 2^c$ tokens per window:

$$\begin{aligned} \hat{\mathbf{x}}_\mathbf{c} &= \text{Conv}_{3\times3}(\mathbf{x}), \\ \hat{\mathbf{x}}_\mathbf{ct} &= \text{AvgPool}_{H^2 \rightarrow n^2 L}(\hat{\mathbf{x}}_\mathbf{c}), \end{aligned} \tag{3}$$

where $\text{Conv}_{3\times3}$ represents efficient positional encoding inspired by [13] and used in Twins [11]. $\hat{\mathbf{x}}_\mathbf{ct}$ and AvgPool denote the carrier tokens and feature pooling operation, respectively; c is set to 1, but can be changed to control latency. The current approach with conv+pooling gives flexibility with the image size. These pooled tokens represent a summary of their respective local windows, we set $L << k$. The procedure of CT initialization is performed only once for every resolution stage. Note that every local window $\hat{\mathbf{x}}_\mathbf{l}$ has unique set of carrier tokens, $\hat{\mathbf{x}}_{\mathbf{ct},\mathbf{l}}$, such that $\hat{\mathbf{x}}_\mathbf{ct} = \{\hat{\mathbf{x}}_{\mathbf{ct},\mathbf{l}}\}_{\mathbf{l}=0}^{n}$.

In every HAT block, CTs undergo the attention procedure:

$$\begin{aligned} \hat{\mathbf{x}}_\mathbf{ct} &= \hat{\mathbf{x}}_\mathbf{ct} + \gamma_1 \cdot \text{MHSA}(\text{LN}(\hat{\mathbf{x}}_\mathbf{ct})), \\ \hat{\mathbf{x}}_\mathbf{ct} &= \hat{\mathbf{x}}_\mathbf{ct} + \gamma_2 \cdot \text{MLP}_{d\rightarrow4d\rightarrow d}(\text{LN}(\hat{\mathbf{x}}_\mathbf{ct})), \end{aligned} \tag{4}$$

where LN represents layer normalization [1], MHSA represents multi-head self attention [56], γ is a learnable per-channel scale multiplier [54], $\text{MLP}_{d\rightarrow4d\rightarrow d}$ is a 2-layer MLP with GeLU [30] activation function.

Next, in order to model short-long-range spatial information, we compute the interaction between the local and carrier tokens, $\hat{\mathbf{x}}_\mathbf{l}$ and $\hat{\mathbf{x}}_{\mathbf{ct},\mathbf{l}}$, respectively. At first, local features and CTs are concatenated. Each local window only has access to its corresponding CTs:

$$\hat{\mathbf{x}}_\mathbf{w} = \text{Concat}(\hat{\mathbf{x}}_\mathbf{l}, \hat{\mathbf{x}}_{\mathbf{ct},\mathbf{l}}). \tag{5}$$

These tokens undergo another set of attention procedure:

$$\begin{aligned} \hat{\mathbf{x}}_\mathbf{w} &= \hat{\mathbf{x}}_\mathbf{w} + \gamma_1 \cdot \text{MHSA}(\text{LN}(\hat{\mathbf{x}}_\mathbf{w})), \\ \hat{\mathbf{x}}_\mathbf{w} &= \hat{\mathbf{x}}_\mathbf{w} + \gamma_2 \cdot \text{MLP}_{d\rightarrow4d\rightarrow d}(\text{LN}(\hat{\mathbf{x}}_\mathbf{w})). \end{aligned} \tag{6}$$

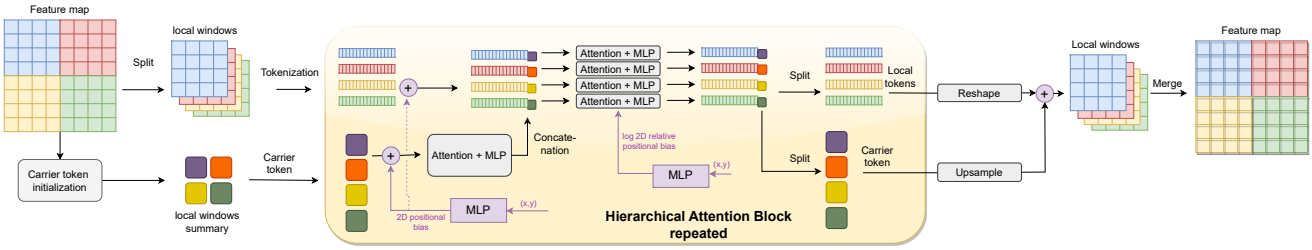
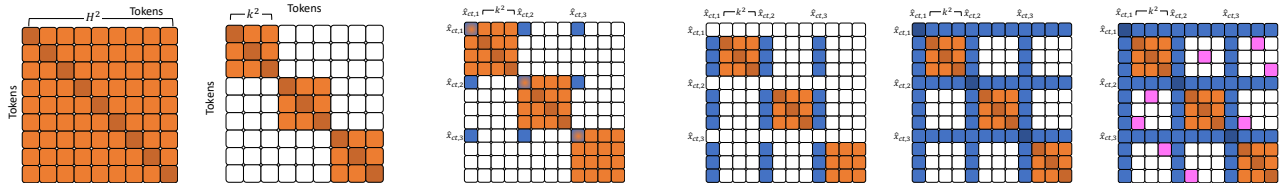


Figure 4. Proposed Hierarchical Attention block. Carrier tokens (CT) learn a summary of each local window and facilitate global information exchange between local windows. Local window tokens only have access to a dedicated subset of CT for efficient attention. CT undergo full self-attention to enable cross-window attention. "Attention" stands for MHSA [56], MLP for multi-layer perceptron. Best viewed in color.



(a) Full attention (b) Windowed attention (c) **Hierarchical Attention (ours)** (d) Twins [11] (e) LongFormer [2] (f) BigBird [74]

Figure 5. Attention map comparison for a feature map of size $H \times H \times d$. ▢ - no attention, ■ - normal token attention, ■ - carrier token attention, ■ - random token attention. Full attention (a) has complexity of $O(H^4d)$, windowed attention significantly reduces it to $O(k^2H^2d)$ but lacks global context. The proposed attention mechanism allows for context to be passed on by carrier tokens. It is implemented through 2 dense attentions on (i) window attention and (ii) carrier token attention. Twins [11] uses heuristics to compute summarization tokens, they are used in all windowed attention and have higher computational complexity while lacking cross summarization token interaction. LongFormer [2] and BigBird [74] could be adapted (from 1D) to 2D as shown in (e) and (f), they have a more dense attention map. Best viewed in color.

Finally, tokens are further split back and used in the subsequent hierarchical attention layers:

$$\hat{\mathbf{x}}_{\mathbf{l}}, \hat{\mathbf{x}}_{\mathbf{ct,l}} = \text{Split}(\hat{\mathbf{x}}_{\mathbf{w}}), \tag{7}$$

Procedures described in Equations 4-7 are iteratively applied for a number of layers in the stage. To further facilitate long-shot-range interaction, we perform *global information propagation*, similar to the one in [44] in the end of the stage. Finally, the output of the stage is computed as:

$$\mathbf{x} = \text{Upsample}_{n^2L \rightarrow H^2}(\hat{\mathbf{x}}_{\mathbf{ct,l}}) + \text{Merge}_{n^2k^2 \rightarrow H^2}(\hat{\mathbf{x}}_{\mathbf{l}}) \tag{8}$$

MHSAs performed in Eq. 4 and 6 are token position invariant, however, the location of features in the spatial dimension are clearly informative. To address this, we first add absolute positional bias directly to CTs and local window tokens. We are inspired by SwinV2 [37] and employ a 2-layer MLP to embed absolute 2D token location into feature dimension. Then, to facilitate image-like locality inductive bias we enhance the attention with log space relative positional bias from SwinV2 [37] (2-layer MLP). It ensures that the relative position of tokens contribute to shared attention patterns. This approach yields flexibility regarding image size, as the positional encoding is interpolated by the MLP, and hence a trained model can be applied to any input resolution.

An attention map comparison between efficient global-local self attention is shown in Fig. 5. The proposed hierarchical attention splits full attention into local and sub-global, both compressible to 2 dense attentions. Carrier tokens participate in both attentions and facilitate information exchange.

Complexity Analysis of HAT The key features of the efficiency of our approach are (i) separation of attentions and (ii) local windows only have access to their CTs. The complexity of the most conventional and popular full attention is $O(H^4d)$. Partitioning the feature size into windows of size k, and running the attention, simplifies the attention to $O(k^2H^2d)$ as proposed in [38]. It is well known that such windowed attention is more efficient but lacks global feature interaction. Our approach takes this one step further and is based on carrier tokens that summarize and interact over the entire feature map, to remedy for missing global communication. Given L total carrier tokens per window, local window complexity is $O((k^2 + L)H^2d)$. Local (windowed) attention is followed by attention on carrier tokens with complexity $O((\frac{H^2}{k^2}L)^2d)$. The total cost of both attentions is $O(k^2H^2d + LH^2d + \frac{H^4}{k^4}L^2d)$.

An orthogonal approach for multilevel attention is to provide access to subsampled global information inside local attention. For example, Twins [11] subsamples global feature map and uses it as key and value for local window attention. It has a complexity of $O(k^2H^2d + \frac{H^4}{k^2}d)$ (from

Table 1. Comparison of classification benchmarks on **ImageNet-1K** dataset [15]. Image throughput is measured on A100 GPUs with batch size of 128.

Model	Image Size (Px)	#Param (M)	FLOPs (G)	Throughput (Img/Sec)	Top-1 (%)
Conv-Based					
ConvNeXt-T [39]	224	28.6	4.5	3196	82.0
ConvNeXt-S [39]	224	50.2	8.7	2008	83.1
ConvNeXt-B [39]	224	88.6	15.4	1485	83.8
RegNetY-040 [46]	288	20.6	6.6	3227	83.0
ResNetV2-101 [59]	224	44.5	7.8	4019	82.0
EfficientNetV2-S [51]	384	21.5	8.0	1735	83.9
Transformer-Based					
Swin-T [38]	224	28.3	4.4	2758	81.3
Swin-S [38]	224	49.6	8.5	1720	83.2
SwinV2-T [37]	256	28.3	4.4	1674	81.8
SwinV2-S [37]	256	49.7	8.5	1043	83.8
SwinV2-B [37]	256	87.9	15.1	535	84.6
TNT-S [25]	224	23.8	4.8	1478	81.5
Twins-S [11]	224	24.1	2.8	3596	81.7
Twins-B [11]	224	56.1	8.3	1926	83.1
Twins-L [11]	224	99.3	14.8	1439	83.7
DeiT-B [52]	224	86.6	16.9	2035	82.0
DeiT3-L	224	304.4	59.7	535	84.8
PoolFormer-M58 [71]	224	73.5	11.6	884	82.4
Hybrid					
CoaT-Lite-S [64]	224	19.8	4.1	2269	82.3
CrossViT-B [5]	240	105.0	20.1	1321	82.2
Visformer-S [9]	224	40.2	4.8	3676	82.1
EdgeViT-S [44]	224	13.1	1.9	4254	81.0
EfficientFormer-L7 [33]	224	82.2	10.2	1359	83.4
MaxViT-B [55]	224	120.0	23.4	507	84.9
MaxViT-L [55]	224	212.0	43.9	376	85.1
FasterViT					
FasterViT-0	224	31.4	3.3	**5802**	**82.1**
FasterViT-1	224	53.4	5.3	**4188**	**83.2**
FasterViT-2	224	75.9	8.7	**3161**	**84.2**
FasterViT-3	224	159.5	18.2	**1780**	**84.9**
FasterViT-4	224	424.6	36.6	**849**	**85.4**
FasterViT-5	224	957.5	113.0	**449**	**85.6**
FasterViT-6	224	1360.0	142.0	**352**	**85.8**

the paper). Under the same size of the local window (k), and H, we can get the difference of $O(L + \frac{H^2L^2}{k^4})$ for HAT and $O(\frac{H^2}{k^2})$ for Twins. HAT gets more efficient with higher resolution, for example, for $H = 32$, $k = 8$, with $L = 4$ we get $O(8)$ for HAT, whereas $O(16)$ for Twins.

4. Experiments

4.1. Training Settings

Image Classification We employ the ImageNet-1K dataset [15] for classification that includes 1.2M and 50K training and validation images. The dataset has 1000 categories and we report the performance in terms of top-1 accuracy. In addition, we use ImageNet-21K dataset which has 14M images with 21841 classes for pretraining. We train all FasterViT models by using LAMB optimizer [70] optimizer for 300 epochs with a learning rate of 5e-3 and a total batch size of 4096 using 32 A100 GPUs. For data augmentation, we follow same strategies as in previous efforts [39, 38]. We also use Exponential Moving Average (EMA) which often improves the performance. Further details on training settings can be found in the appendix. For pre-training on ImageNet-21K, we train the models for 90 epochs with a learning rate of 4e-3. In addition, we fine-tune the models for 60 epochs with a learning rate of 7e-5.

Table 2. **ImageNet-21K** pretrained classification benchmarks on **ImageNet-1K** dataset [15]. Image throughput is measured on A100 GPUs with batch size of 128. ‡ denotes models that are pre-trained on ImageNet-21K dataset.

Model	Image Size (Px)	#Param (M)	FLOPs (G)	Throughput (Img/Sec)	Top-1 (%)
ViT-L/16‡ [38]	384	307.0	190.7	149	85.2
Swin-L‡ [38]	224	197.0	34.5	787	86.3
Swin-L‡ [38]	384	197.0	103.9	206	87.3
ConvNeXt-L‡ [39]	224	198.0	34.4	508	86.6
ConvNeXt-L‡ [39]	384	198.0	101.0	172	87.5
FasterViT-4‡	224	424.6	36.6	**849**	**86.6**
FasterViT-4‡	384	424.6	119.2	**281**	**87.5**

Table 3. Object detection and instance segmentation benchmarks using Cascade Mask R-CNN [28] on **MS COCO** dataset [36]. All models employ 3× schedule. All model statistics are reported using a input test resolution of 1280 × 800.

Backbone	Throu. im/sec	AP^{box} Box	50	75	AP^{mask} Mask	50	75
Swin-T [38]	161	50.4	69.2	54.7	43.7	66.6	47.3
ConvNeXt-T [39]	166	50.4	69.1	54.8	43.7	66.5	47.3
DeiT-Small/16 [52]	269	48.0	67.2	51.7	41.4	64.2	44.3
FasterViT-2	**287**	**52.1**	**71.0**	**56.6**	**45.2**	**68.4**	**49.0**
Swin-S [38]	119	51.9	70.7	56.3	45.0	68.2	48.8
X101-32 [62]	124	48.1	66.5	52.4	41.6	63.9	45.2
ConvNeXt-S [39]	128	51.9	70.8	56.5	45.0	68.4	49.1
FasterViT-3	**159**	**52.4**	**71.1**	**56.7**	**45.4**	**68.7**	**49.3**
X101-64 [62]	86	48.3	66.4	52.3	41.7	64.0	45.1
Swin-B [38]	90	51.9	70.5	56.4	45.0	68.1	48.9
ConvNeXt-B [39]	101	52.7	71.3	57.2	45.6	68.9	49.5
FasterViT-4	**117**	**52.9**	**71.6**	**57.7**	**45.8**	**69.1**	**49.8**

Table 4. **MS COCO** dataset [36] object detection results with DINO [75] model. ‡ denotes models that are pre-trained on ImageNet-21K dataset.

Backbone	Model	Epochs	FLOPs (G)	Throughput	AP^{box}
Swin-L‡ [38]	HTC++ [6]	72	1470	-	57.1
Swin-L‡ [38]	DINO [75]	36	1285	71	58.5
FasterViT-4‡	DINO [75]	36	1364	**84**	**58.7**

Detection and Segmentation We used the MS COCO dataset [36] to finetune a Cascade Mask-RCNN network [28] with pretrained FasterViT backbones. For this purpose, we trained all models with AdamW [40] optimizer with an initial learning rate of 1e-4, a 3 × schedule, weight decay of 5e-2 and a total batch size of 16 on 8 A100 GPUs.

Table 5. Semantic segmentation benchmarks **ADE20K** dataset [78] with UPerNet [60] and pre-trained FasterViT backbones. All model statistics are reported using a input test resolution of 512×512. Throughput is measured in image/sec.

Model	Throughput	FLOPs (G)	IoU(ss/ms)
Swin-T [38]	350	945	44.5/45.8
ConvNeXt-T [39]	363	939	- /46.7
FasterViT-2	**377**	974	**47.2/48.4**
Twins-SVT-B [11]	204	-	47.7/48.9
Swin-S [38]	219	1038	47.6/49.5
ConvNeXt-S [39]	234	1027	- /49.6
FasterViT-3	**254**	1076	**48.7/49.7**
Twins-SVT-L [11]	164	-	48.8/50.2
Swin-B [38]	172	1188	48.1/49.7
ConvNeXt-B [39]	189	1170	- /49.9
FasterViT-4	**202**	1290	**49.1/50.3**

Semantic Segmentation For semantic segmentation, we employed ADE20K dataset [78] to finetune an UperNet network [60] with pre-trained FasterViT backbones. Specifically, we trained all models with Adam-W [40] optimizer and by using a learning rate of 6e-5, weight decay of 1e-2 and total batch size of 16 on 8 A100 GPUs.

5. Results

5.1. Image Classification

In Table 1, we demonstrate a quantitative comparison between the performance of FasterViT models and a variety of different hybrid, conv and Transformer-based networks on ImageNet-1K dataset. Comparing to Conv-based architectures, we achieve higher accuracy under the same throughput, for example, we outperform ConvNeXt-T by 2.2%. Considering the accuracy and throughput trade-off, FasterViT models are significantly faster than Transformer-based models such as the family of Swin Transformers [38, 37]. Furthermore, compared to hybrid models, such as the recent EfficientFormer [33] and MaxViT [55] models, FasterViT on average has a higher throughput while achieving a better ImageNet top-1 performance. The trend of latency-accuracy Pareto front holds even for post training model optimization techniques such as TensorRT (see appendix).

In order to validate the scalability of the proposed model, we pre-trained FasterViT-4 on ImageNet-21K dataset and fine-tuned it on various image resolutions on ImageNet-1K dataset. In general, FasterViT-4 has a better accuracy-throughput trade-off compared to other counterparts. As shown in Table 2, FasterViT-4 outperforms ViT-L/16 by a significant margin of +2.3% on 384^2 resolution and +15.43% higher throughput. In addition, FasterViT-4 outperforms Swin-L on both 224^2 and 384^2 resolutions by +0.3% and +0.2% while having 12.07% and 36.40% higher throughput, respectively. FasterViT has 73.62% and 63.37% higher throughput compared to ConvNeXt-L on 224^2 and 384^2 resolutions respectively, while achieving the same Top-1 accuracy benchmarks.

Table 6. Ablation study on the effectiveness of HAT compared to EdgeViT [44] and Twins [11] self-attention mechanisms. All attention blocks are replaced with the indicated attention type.

Model	Attention	FLOPs (G)	Throughput (Img/Sec)	Top-1 (%)
FasterViT-0	Twins [11]	3.0	6896	80.8
FasterViT-0	EdgeViT [44]	3.2	5928	81.0
FasterViT-0	**HAT**	3.3	5802	**82.2**
FasterViT-1	Twins [11]	4.7	4949	82.1
FasterViT-1	EdgeViT [44]	4.8	4188	82.5
FasterViT-1	**HAT**	5.3	4344	**83.2**
FasterViT-2	Twins [11]	8.0	3668	82.9
FasterViT-2	EdgeViT [44]	8.5	3127	83.4
FasterViT-2	**HAT**	8.7	3161	**84.1**

5.2. Object Detection and Instance Segmentation

In Table 3, we present object detection and instance segmentation benchmarks on MS COCO dataset [36] with Cascade Mask R-CNN [28] network. We observe that FasterViT models have better accuracy-throughput trade-off when compared to other counterparts. Specifically, FasterViT-4 outperforms ConvNeXt-B and Swin-B by +0.2 and +1.0 in terms of box AP and +0.3 and +1.0 in terms of mask AP, while being 15% and 30% faster in terms of throughput, respectively. Similar trend are observed for other models variants. In addition, as shown in In Table 3, we conduct additional object detection experiments with FasterViT-4 ImageNet-21K pre-trained backbone and the state-of-the-art DINO [75] model and achieve a high detection accuracy of 58.7 box AP. Hence, this validates the effectiveness of FasterViT as a backbone with more sophisticated and state-of-the-art models.

5.3. Semantic Segmentation

In Table 5, we present the semantic segmentation benchmarks with UPerNet [60] network for experiments conducted on ADE20K dataset [78]. Similar to previous tasks, FasterViT models benefit from a better performance-throughput trade-off. Specifically, FasterViT-4 outperforms Swin-B by +1.0 and +0.7 for single and multi scale inference in terms of mIoU, respectively while having 16.94% higher throughput. Similarly, FasterViT-4 has a 7.01% higher throughput and achieves +0.4 higher mIoU for multi scale inference compared to ConvNeXt-B.

6. Ablation

6.1. Component-wise study

Table 9 shows per component ablation. Two settings are considered: (i) when the model is trained without the component, (ii) when the component is disabled after the model is trained. The first shows if the model can operate

Table 7. Quantitative comparison between higher resolution finetuning of FasterViT and Swin Transformer V2 networks. FasterViT is more accurate on average by 0.9%, and faster by 2x.

	Pretrain		Finetune					
	W8, I256		W12, I384		W16, I512		W24, I768	
Model	acc	im/s	acc	im/s	acc	im/s	acc	im/s
SwinV2-T [37]	81.8	1674	83.2	573	83.8	168	84.2	72
SwinV2-S [37]	83.7	633	84.8	338	85.4	153	-	-
FasterViT-2	**84.3**	**2500**	**85.3**	**984**	**85.5**	**489**	**85.6**	**155**
SwinV2-B [37]	84.2	499	85.1	251	85.6	115	-	-
FasterViT-4 256	**85.3**	**653**	**86.0**	**254**	**86.1**	**133**	**86.0**	**44**

Table 8. Ablation study on the effectiveness of HAT as a plug-and-play module with Swin-T model for various CV tasks.

	ImageNet	COCO		ADE20k
	top-1	AP^{box}	AP^{mask}	mIoU
Swin-T	81.3	50.4	43.7	44.5
Swin-T + HAT	**81.7**	**50.9**	**44.3**	**45.4**

Table 9. Ablation study on the effectiveness of HAT components in FasterViT-2 as measured in Top-1 change on ImageNet-1K [15].

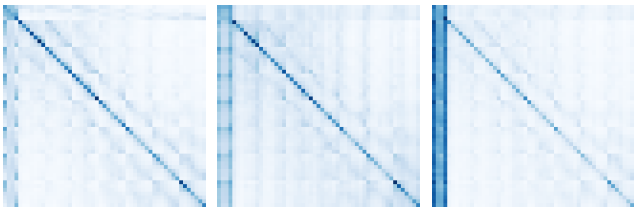
Ablation	Trained from scratch	Post training removal	Throughput ratio
HAT block	-0.24%	-1.49%	1.08
CT attention	-0.13%	-3.85%	1.00
Attention Bias	-0.31%	-8.90%	1.00
CT propagation	-0.16%	-	0.93
1D pos bias	-0.07%	-24.85%	1.00
CT initialization	-0.05%	-0.48%	1.00
Window 14×14	+0.10%	-	0.90

well without the component, while the second cases shows if the components is used in the final model.

We observe that changing the window resolution to 14×14 in the 3rd stage (effectively removing HAT by have a full global window) improves the model accuracy by $+0.1\%$ while scarifying 10% of throughput. Even though this setup shows better accuracy, it does not scale to high resolution, and HAT is required. Removing the HAT block from the architecture results in -0.24% accuracy drop for re-trained model and -1.49% for post training study at the benefit of 8% throughtput improvement. CT attention is another block of high importance, resulting in -3.85% post training removal. Attention bias is an important component of our system, resulting in -0.31% drop in the re-training scenario. Removing CT propagation, results in the requirement to pool and propagate features at every layer (similar to EdgeViT), that costs 7% of total inference and in lower accuracy -0.16%. CT initialization is important to the network, as accuracy drops by -0.48% in post-training removal. Removing all components and having only CNN plus windowed vanilla transformer results in -0.46%.

6.2. Attention Maps

In Fig. 6, we have illustrated the full attention maps of stage 3 layers for different FasterViT model variants. Specifically, the attention maps have a resolution of 53×53 consisting of a concatenation of 4×4 carrier tokens and 49×49 local window-based attention. The carrier tokens are in the top left position of each attention map. We observe that all local tokens attend to the carrier tokens in addition to their own local attention. Please see the appendix for more visualizations.



(a) FasterViT-2 (b) FasterViT-3 (c) FasterViT-4

Figure 6. (a) FasterViT-2. (b) FasterViT-3. (c) FasterViT-4. Full attention map visualizations of stage 3 for FasterViT model variants.

6.3. Attention Alternatives

As shown in Table 6, we performed a comprehensive ablation study to validate the effectiveness of HAT by replacing all attention layers with attention mechanisms in EdgeViT [44] and Twins [11] in the 3rs and 4th stages. For all model variants, FasterViT models with HAT achieve a better accuracy, sometimes by a significant margin. Twins achieves a higher throughput due to its small kernel size (*i.e.* $k = 2$), however, this significantly limits its accuracy. The better performance of HAT is attributed to its learnable information aggregation/propagation via CTs, and direct access to dedicated CTs in windowed attention.

Plug-and-Play HAT. We employed HAT as a plug-and-play module with Swin-T model Table 8. This change results in +0.9 and +0.4% improvement in terms of mIoU and Top-1 accuracy on ImageNet classification and ADE20K segmentation tasks. In addition, improvements on MS COCO by +0.5 box AP and +0.6 mask AP on object detection and instance segmentation tasks, respectively. These results indicate the viability of HAT as a standalone self-attention mechanism.

7. Conclusion

In this work, we have presented a novel hybrid model, denoted as FasterViT, which achieves SOTA Pareto-front in terms of ImageNet Top-1 accuracy and throughput. We introduced an novel Hierarchical Attention (HAT) which computes cross-window interactions at reduced computational cost. We have extensively validated the effectiveness of FasterViT in downstream tasks such as object detection, instance segmentation and semantic segmentation. Our benchmarks demonstrate better accuracy-throughput trade-off in comparison to counterpart models such as ConvNeXt and Swin Transformer. We have also demonstrated that HAT can be used as a plug-and-play self-attention module and improve the accuracy and throughput for various tasks.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4, 14, 15

[2] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 3, 5

[3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. Hydra attention: Efficient attention with many heads. *arXiv preprint arXiv:2209.07484*, 2022. 3

[4] Han Cai, Chuang Gan, and Song Han. Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition. *arXiv preprint arXiv:2205.14756*, 2022. 3

[5] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021. 3, 6

[6] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4974–4983, 2019. 6

[7] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. *arXiv preprint arXiv:2107.00651*, 2021. 3

[8] Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *arXiv preprint arXiv:2106.04533*, 2021. 3

[9] Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 589–598, 2021. 3, 6, 14

[10] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 1

[11] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 3, 4, 5, 6, 7, 8, 14

[12] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 3

[13] Xiangxiang Chu, Bo Zhang, Zhi Tian, Xiaolin Wei, and Huaxia Xia. Do we really need explicit position encodings for vision transformers? *CoRR*, abs/2102.10882, 2021. 4

[14] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021. 3

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6, 8, 12, 13

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2019. 3

[17] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, 2022. 1, 3

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 1, 2

[19] Jiawei Du, Zhou Daquan, Jiashi Feng, Vincent Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. In *Advances in Neural Information Processing Systems*. 15

[20] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021. 2

[21] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34:26183–26197, 2021. 1

[22] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2021. 3

[23] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020. 15

[24] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12259–12269, 2021. 2

[25] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021. 6

[26] Ali Hatamizadeh, Hongxu Yin, Jan Kautz, and Pavlo Molchanov. Global context vision transformers. *arXiv preprint arXiv:2206.09959*, 2022. 3

[27] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1

[28] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6, 7

[29] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 12

[30] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[31] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 12

[32] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 4

[33] Yanyu Li, Geng Yuan, Yang Wen, Eric Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *arXiv preprint arXiv:2206.01191*, 2022. 3, 6, 7

[34] Youwei Liang, Chongjian GE, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EVit: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. 3

[35] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1, 2

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 6, 7

[37] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022. 2, 3, 5, 6, 7, 8, 13, 14, 15

[38] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 2, 3, 4, 5, 6, 7, 12, 14

[39] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 6, 7, 12

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6, 7, 15

[41] Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: softmax-free transformer with linear complexity. *Advances in Neural Information Processing Systems*, 34:21297–21309, 2021. 3

[42] Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers. *arXiv preprint arXiv:2110.03860*, 2021. 3

[43] Pavlo Molchanov, Jimmy Hall, Hongxu Yin, Jan Kautz, Nicolo Fusi, and Arash Vahdat. Lana: latency aware network acceleration. In *European Conference on Computer Vision*, pages 137–156. Springer, 2022. 3

[44] Junting Pan, Adrian Bulat, Fuwen Tan, Xiatian Zhu, Lukasz Dudziak, Hongsheng Li, Georgios Tzimiropoulos, and Brais Martinez. Edgevits: Competing light-weight cnns on mobile devices with vision transformers. In *ECCV*, 2022. 3, 5, 6, 7, 8, 14

[45] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2071–2081, 2022. 1

[46] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020. 6

[47] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34, 2021. 1

[48] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021. 3

[49] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019. 12

[50] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019. 3

[51] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021. 6

[52] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1, 2, 6

[53] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 516–533. Springer, 2022. 15

[54] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers, 2021. 4

[55] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *Computer Vision–ECCV*

2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV, pages 459–479. Springer, 2022. 6, 7

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 4, 5

[57] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021. 2

[58] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 2

[59] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021. 6

[60] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 7

[61] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 1

[62] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 6

[63] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. 1

[64] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9981–9990, 2021. 6

[65] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems*, 34:28522–28535, 2021. 1

[66] Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2964–2972, 2022. 3

[67] Huanrui Yang, Hongxu Yin, Pavlo Molchanov, Hai Li, and Jan Kautz. Nvit: Vision transformer compression and parameter redistribution. *arXiv preprint arXiv:2110.04869*, 2021. 3

[68] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal attention for long-range interactions in vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[69] Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[70] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019. 6, 15

[71] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10819–10829, 2022. 6

[72] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token ViT: Training vision transformers from scratch on imagenet. In *ICCV*, 2021. 3

[73] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3

[74] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*, 2020. 3, 5

[75] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 6, 7

[76] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. *arXiv preprint arXiv:2103.15358*, 2021. 3

[77] Zixiao Zhang, Xiaoqiang Lu, Guojin Cao, Yuting Yang, Licheng Jiao, and Fang Liu. Vit-yolo: Transformer-based yolo for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2799–2808, 2021. 1

[78] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 7

Appendix

A. Robustness Analysis

In this section, we analyze the robustness of FasterViT models on different datasets. We test FasterViT model variants on ImageNet-A [31], ImageNet-R [29] and ImageNetV2 [49] datasets. In addition, we did not perform any fine-tuning and simply employed the pre-trained ImageNet-1K [15] weights for each model. As shown in Table S.1, FasterViT demonstrates promising robustness performance on various datasets for each model variant. Specifically, FasterViT-3 outperforms comparable models such as ConvNeXt-B and Swin-B [39] by +7.5% and +8.4% on ImageNet-A [31], +0.6% and +5.3% on ImageNet-R [29] and +1.3% and +2.7% on ImageNetV2 [49], respectively. For larger models, FasterViT-4 outperforms ConvNeXt-L [39] by +7.9%, +2.6% and +1.5% on ImageNet-A [31], ImageNet-R [29] and ImageNetV2 [49], respectively, hence validating the effectiveness of the proposed model in various benchmarks. Similar trends can be observed for smaller models.

Table S.1. Robustness analysis of **ImageNet-1K** [15] pretrained FasterViT models on ImageNet-A [31], ImageNet-R [29] and ImageNetV2 [49] datasets.

Model	Size (Px)	#Param (M)	FLOPs (G)	Throughput (Img/Sec)	Clean (%)	A (%)	R (%)	V2 (%)
FasterViT-0	224	31.4	3.3	**5802**	**82.1**	**23.9**	**45.9**	**70.9**
FasterViT-1	224	53.4	5.3	**4188**	**83.2**	**31.2**	**47.5**	**72.6**
Swin-T [38]	224	28.3	4.4	2758	81.3	21.6	41.3	69.7
ConvNeXt-T [39]	224	28.6	4.5	3196	82.0	24.2	47.2	71.0
ConvNeXt-S [39]	224	50.2	8.7	2008	83.1	31.3	49.5	72.4
FasterViT-2	224	75.9	8.7	**3161**	**84.2**	**38.2**	**49.6**	**73.7**
Swin-S [38]	224	49.6	8.5	1720	83.2	32.5	44.7	72.1
Swin-B [38]	224	87.8	15.4	1232	83.4	35.8	46.6	72.3
ConvNeXt-B [39]	224	88.6	15.4	1485	83.8	36.7	51.3	73.7
FasterViT-3	224	159.5	18.2	**1780**	**84.9**	**44.2**	**51.9**	**75.0**
ConvNeXt-L [39]	224	198.0	34.4	508	84.3	41.1	53.4	74.2
FasterViT-4	224	424.6	36.6	**849**	**85.4**	**49.0**	**56.0**	**75.7**
FasterViT-5	224	975.5	113.0	**449**	**85.6**	**52.7**	**56.9**	**76.0**
FasterViT-6	224	1360.0	142.0	**352**	**85.8**	**53.7**	**57.1**	**76.1**

B. Ablation

B.1. FasterViT Component-wise Study

Table S.2 shows per component ablation. Two settings are considered: (i) when the model is trained without the component, (ii) when the component is disabled after the model is trained. The first shows if the model can operate well without the component, while the second cases shows if the components is used in the final model.

We observe that changing the window resolution to 14×14 in the 3rd stage (effectively removing HAT by have a full global window) improves the model accuracy by $+20\%$ while scarifying 10% of throughput. Even though this setup shows better accuracy, it does not scale to high resolution, and HAT is required.

Removing the HAT block from the architecture results in -0.24% accuracy drop for re-trained model and -1.49% for post training study at the benefit of 8% throughtput improvement. CT attention is another block of high importance, resulting in -3.85% post training removal. Attention bias is an important component of our system, resulting in -0.31% drop in the re-training scenario. Removing CT propagation, results in the requirement to pool and propagate features at every layer (similar to EdgeViT), that costs 7% of total inference and in lower accuracy -0.16%. CT initialization is important to the network, as accuracy drops by -0.48% in post-training removal. Removing all components and having only CNN plus windowed vanilla transformer results in -0.46%.

Table S.2. Ablation study on the effectiveness of HAT components in FasterViT-2 as measured in Top-1 change on ImageNet-1K [15].

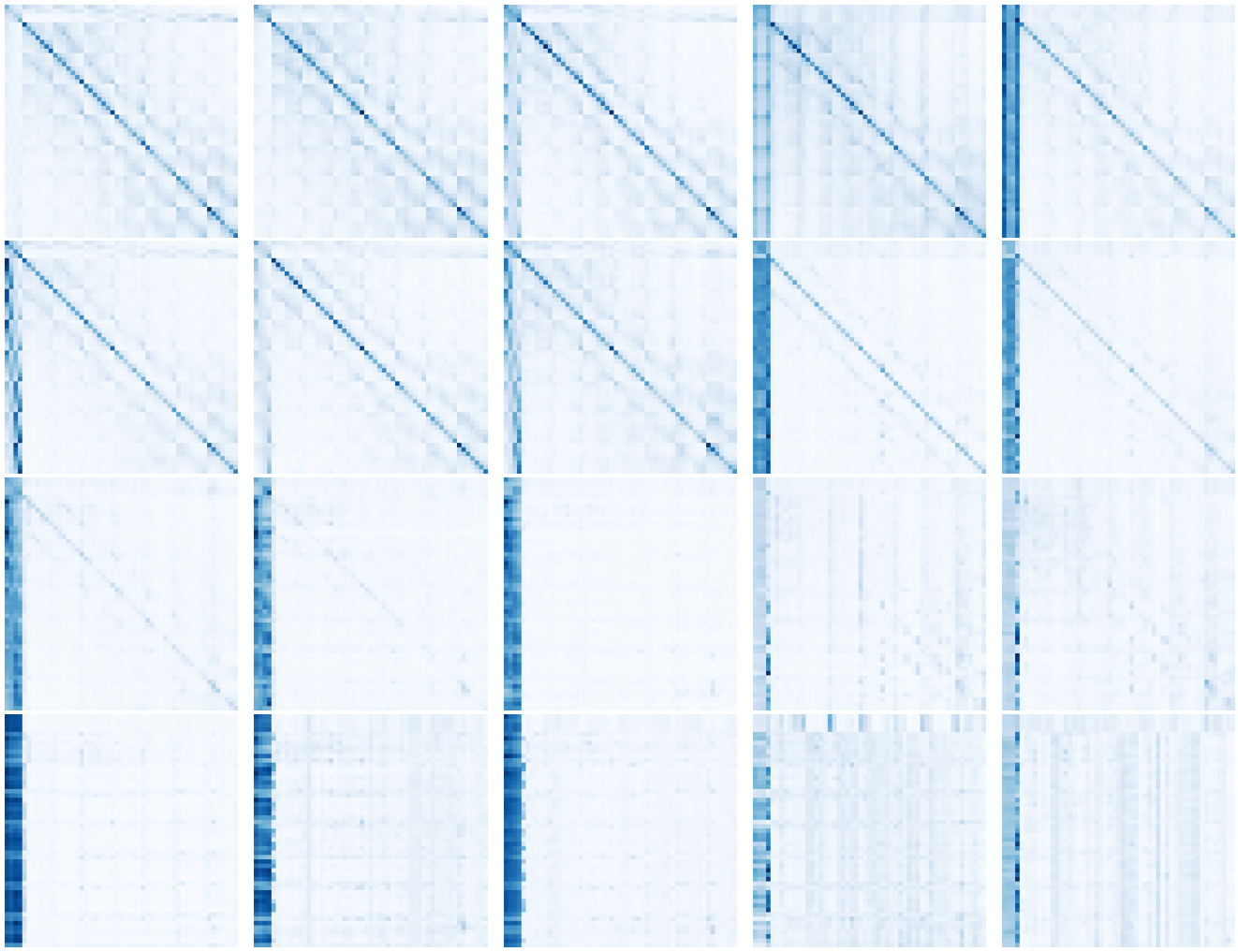
Ablation	Trained from scratch	Post training removal	Throughput ratio
HAT block	-0.24%	-1.49%	1.08
CT attention	-0.13%	-3.85%	1.00
Attention Bias	-0.31%	-8.90%	1.00
CT propagation	-0.16%	-	0.93
1D pos bias	-0.07%	-24.85%	1.00
CT initialization	-0.05%	-0.48%	1.00
Window 14×14	+0.20%	-	0.90

Table S.3. Ablation study of the number of carrier tokens. The numbers after w and c indicate the window size and the carrier token window size, respectively.

224 res						
Config	w14c0	w7c7	w7c6	w7c5	w7c2	w7c1
Acc	84.26	84.92	84.41	84.28	84.16	83.96
Latency ratio	0.9	0.47	0.57	0.67	1.0	1.05
256 res						
Config	w16c0	w8c4	w8c3	w8c2		
Acc	84.65	84.7	84.51	84.4		
Latency ratio	0.95	0.87	0.93	1.0		

B.2. Number of carrier tokens

The proposed HAT can be controlled by the number of carrier tokens. We ablate the carrier token window size in Table S.3. It should be noted, that the number of model parameters does not change with varying the number of CT, however, the latency does.



(a) FasterViT-0 (b) FasterViT-1 (c) FasterViT-2 (d) FasterViT-3 (d) FasterViT-4

Figure S.1. (a) FasterViT-0. (b) FasterViT-1. (c) FasterViT-2. (d) FasterViT-3 (e) FasterViT-4. Full attention map visualizations of stage 3 for FasterViT model variants. From top to bottom, we visualize attention maps of first to last layers with an interval of a quarter length of the number of layers in stage 3 for each model. We visualize the attention maps of the same input image for all cases to facilitate comparability.

B.3. SwinV2 Comparison

In the Table 7 we compare the performance of SwinV2 [37] and FasterViT models on large image resolution. The initial model is pretrained with an image resolution of 256^2px for 300 epochs on ImageNet-1K. Then models are fine-tuned on a larger resolution (I) for an 30 epochs with various window sizes (W). Faster-ViT consistently demonstrates a higher image throughput, sometimes by a significant margin compared to Swin Transformer V2 model. Hence validating the effectiveness of the proposed hierarchical attention for high input resolution.

C. Attention Maps

In Fig. S.1, we have illustrated the full attention maps of stage 3 layers for different FasterViT model variants. For this purpose, we use input images of size 224 × 224 × 3 and ImageNet-1K [15] trained FasterViT models. For each model, from the top to the bottom rows, we show the attention maps from the first to the final layer with an interval of a quarter of the total number of layers at stage 3 (*e.g.* layers 1, 4, 9 and 12 for FasterViT-4).

In particular, Stage 3 for this illustration serves an important purpose, since we use local attention windows of 7×7 with input features that have a resolution of 14 × 14. Hence, attention is computed in 4 local regions after window partitioning and 4 carrier tokens are designated to each corresponding window. Each illustrated attention map has a size of size 53×53 consisting of a concatenation of 4×4 carrier tokens and 49×49 local window-based attention. The carrier tokens are shown in in the top left position of each map. We observe that for all models, all tokens will attend

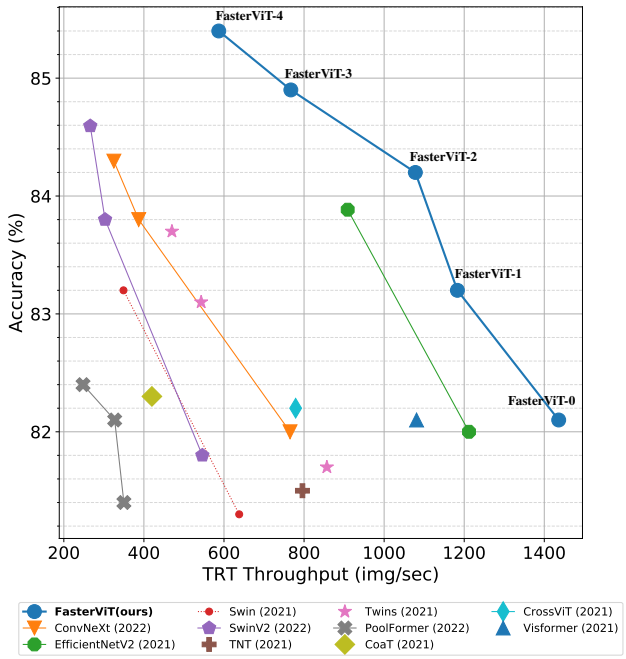


Figure S.2. Comparison of image throughput and ImageNet-1K Top-1 accuracy with TensorRT post-training model optimization. For all models, throughput is measured on A100 GPU with batch size of 1.

to the carrier tokens with different patterns.

For FasterViT-0 and FasterViT-1 models, from the first to the last layers, all tokens transition to attend to the the carrier tokens (*i.e.* vertical bar on the left side). In the last layers, in addition to all tokens attending to the carrier tokens, we see a more global attention pattern, hence showing the cross interaction between different regions.

For FasterViT-2, FasterViT-3 and FasterViT-4 models, starting from the first layers, all tokens attend to both carrier and local tokens. In the last layers however, the attention pattern shifts from local to global. As discussed in this work and also shown in these illustrations, carrier tokens serve an integral role in modeling cross-region interactions and capturing long-range spatial dependencies.

D. TensorRT latency

All throughput numbers and insights presented in the main paper were computed using PyTorch v1.13. In order to demonstrate the scalability with post-training optimization techniques, we compared throughput using the TensoRT (TRT) framework for *batch size 1*, as illustrated in Fig S.2. FasterViT is still considerably faster than other models, making it a good choice to meet various efficient inference design targets.

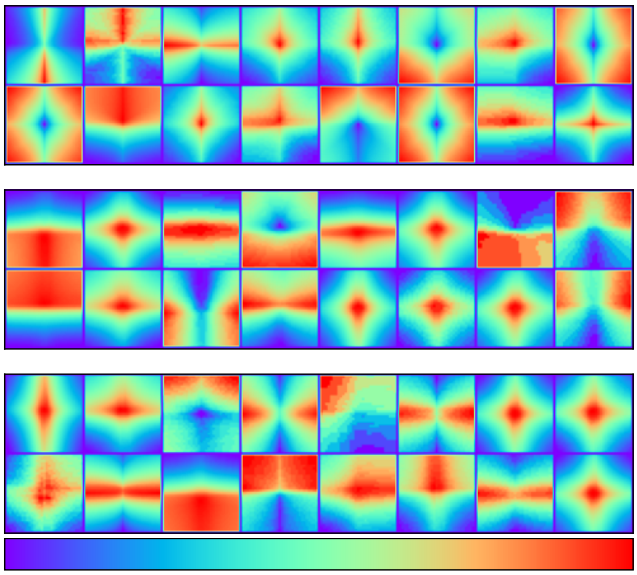


Figure S.3. Learned positional biases for attentions in the 3rd stage of FasterViT-4 model finetuned for 512×512 px. Each kernel corresponds to a bias for a single head in the multi-headed attention. Visualizations demonstrate that the model learns positional dependent features, while also sharing the pattern between pixels.

E. Attention bias

We follow the concept of relative positional bias in the attention from Swin [38]. Particularly, we use the implementation with MLP from SwinV2 [37], where relative coordinate shift in x, y is transformed to the positional bias in the attention via 2-layer network. This allows the model to learn relative position aware kernels, and to introduce image inductive bias. We visualize learned positional biases of the MLP in FasterViT-4 finetuned for 512 with window size of 16×16 pixels in Fig S.3. The visualization shows a diverse set of kernels learned by FasterViT model.

F. Design Insights

Layer normalization [1]. We found it to be critical for transformer blocks (stage 3 and 4). Replacing it with batch normalization leads to accuracy drop of 0.7%. The LN performs cross token normalization and affects cross-channel interaction.

No feature map reshaping. In our architecture, we have removed windowing and de-windowing functions from transformer layers. They are usually used to perform convolutions between layers (like in Twins [11], EdgeViT [44], Visformer [9]), or window shifting (Swin [38], SwinV2 [37]). We perform windowing only once once in stages 3 and 4, and keep data as tokenized with channel last. This leads to throughput improvement of 5% for PyTorch and 10% for TensorRT.

	Output Size (Downs. Rate)	FasterViT-1	FasterViT-2	FasterViT-3	FasterViT-4
Stem	112×112 (2×)	[Conv-BN-ReLU C:32, S:2] × 1	[Conv-BN-ReLU C:64, S:2] × 1	[Conv-BN-ReLU C:64, S:2] × 1	[Conv-BN-ReLU C:64, S:2] × 1
		[Conv-BN-ReLU C:80] × 1	[Conv-BN-ReLU C:96] × 1	[Conv-BN-ReLU C:128] × 1	[Conv-BN-ReLU C:196] × 1
Stage 1	56×56 (4×)	LN-2D, Conv, C:160, S:2	LN-2D, Conv, C:192, S:2	LN-2D, Conv, C:256, S:2	LN-2D, Conv, C:392, S:2
		[ResBlock C:160] ×1,	[ResBlock C:192] × 3,	[ResBlock C:256] × 3,	[ResBlock C:392] × 3,
Stage 2	28×28 (8×)	LN-2D, Conv, C:320, S:2	LN-2D Conv, C:384, S:2	LN-2D, Conv, C:512, S:2	LN-2D, Conv, C:768, S:2
		[ResBlock C:320] × 3,	[ResBlock C:384] × 3,	[ResBlock C:512] × 3,	[ResBlock C:768] × 3,
Stage 3	14×14 (16×)	LN-2D, Conv, C:640, S:2	LN-2D, Conv, C:768, S:2	LN-2D, Conv, C:1024, S:2	LN-2D, Conv, C:1568, S:2
		[HAT C:640, head:8] × 8 ,	[HAT C:768, head:8] × 8,	[HAT C:1024, head:8] × 12,	[HAT C:1568, head:16] × 12,
Stage 4	7×7 (32×)	LN-2D, Conv, C:1280, S:2	LN-2D, Conv, C:1536, S:2	LN-2D, Conv, C:2048, S:2	LN-2D, Conv, C:3136, S:2
		[HAT C:1280, head:16] × 5,	[HAT C:1536, head:16] × 5,	[HAT C:2048, head:16] × 5,	[HAT C:3136, head:32] × 5,

Table S.4. FasterViT architecture configurations. BN and LN-2D denote Batch Normalization and 2D Layer Normalization, respectively. HAT denotes Hierarchical Attention block.

LAMB optimizer [70]. We observed incredible stability of LAMB [70] optimizer for training our biggest models (FasterVIT-3 and FasterViT-4), more widely used AdamW [40] was leading to NaNs for some trainings. We attribute this to joined usage of batch normalization and layer normalization [1] in the same model.

Positional bias. We employ 1D positional bias for local and carrier tokens, as well as 2D relative attention bias by MLP introduced in SwinV2 [37]. For 1D bias we remove *log* scale. This approach yields flexibility to the image size, as positional encoding is interpolated by MLP if resolution change. Those positional biases are quick to compute, however, will block all cores in GPUs until positional biases are computed, and will significantly impact the throughput. To address this, we propose to pre-compute positional biases for a given feature resolution and skip the MLP bottleneck, leading to 6% throughput gain.

Drop-out. We found that conventional drop-out on MLP layers and attention has a negative effect on the final accuracy even for big models that overfit. Stochastic depth is helpful; in contrary to recent trends, we found that a small probability (up to 30%) works better than 65% like in DEIT3 [53]. Better regularization can be achieved by increased weight decay. For example, model 4 with drop-path rate of 50% and weight decay of 0.05 achieves 84.91%, while model 4 with drop-path rate of 30% and weight decay of 0.12 achieves 85.15%.

MESA [19]. It is shown to be useful to prevent overfitting of larger models at little overhead. MESA is a simplified version of SAM [23] that forces optimization to have sharper minima at the convergence, naive implementation slows down training by 2x. In MESA, authors propose to simply apply knowledge distillation loss with respect to the EMA weight computed during training, the training overhead is almost not noticeable. We enable it after 25% of the training, coefficient is set proportionally to the model size in range 0.25 (FasterViT-0)-3.0(FasterViT-4).

Intermediate LN. SwinV2 [37] argues that intermediate LN [1] help to stabilize training of large models, we saw accuracy degradation of this approach.

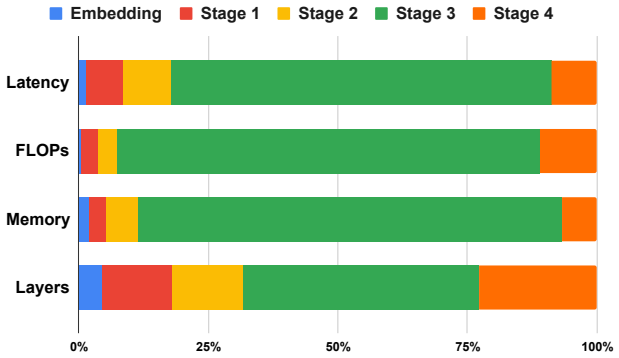


Figure S.4. **FasterViT-2** profiling benchmarks. In FasterViT models, stage 3 (HAT) dominates over all metrics.

G. Architecture Details

In Table S.4, we show the different architecture configurations of the FasterViT model variants.

H. FasterViT Profiling

In Fig. S.4, we provide detailed stage-wise profiling of FasterViT-2 using NVIDIA DLSIM. As expected, stage 3 (HAT) has the highest latency, FLOPs and memory footprint since it is composed of considerably more layers compared to other stages.