Castling-ViT: Compressing Self-Attention via Switching Towards Linear-Angular Attention at Vision Transformer Inference

Haoran You[1,2,†,*], Yunyang Xiong[2,*], Xiaoliang Dai[2], Bichen Wu[2], Peizhao Zhang[2],
Haoqi Fan[2], Peter Vajda[2], Yingyan (Celine) Lin[1]
[1]Georgia Institute of Technology, [2]Meta Research

{haoran.you,celine.lin}@gatech.edu

{yunyang,xiaoliangdai,wbc,stzpz,haoqifan,vajdap}@meta.com

Abstract

Vision Transformers (ViTs) have shown impressive performance but still require a high computation cost as compared to convolutional neural networks (CNNs), one reason is that ViTs' attention measures global similarities and thus has a quadratic complexity with the number of input tokens. Existing efficient ViTs adopt local attention or linear attention, which sacrifice ViTs' capabilities of capturing either global or local context. In this work, we ask an important research question: ==Can ViTs learn both global and local context== *while being more efficient during inference? To this end, we propose a framework called **Castling-ViT**, which* ==trains ViTs using both linear-angular attention and masked softmax-based quadratic attention,== *but* ==then switches to having only linear-angular attention during inference.== *Our Castling-ViT* ==leverages angular kernels to measure the similarities between queries and keys== *via spectral angles. And we further simplify it with two techniques: (1) a novel linear-angular attention mechanism:* ==we decompose the angular kernels into linear terms and high-order residuals, and only keep the linear terms==; *and (2) we adopt two parameterized modules to* ==approximate high-order residuals==: *a* ==depthwise convolution and an auxiliary masked softmax attention== *to help learn global and local information, where the masks for softmax attention are regularized to gradually become zeros and thus incur no overhead during inference. Extensive experiments validate the effectiveness of our Castling-ViT, e.g., achieving up to a **1.8%** higher accuracy or **40%** MACs reduction on classification and **1.2** higher mAP on detection under comparable FLOPs, as compared to ViTs with vanilla softmax-based attentions. Project page is available at here.*

1. Introduction

Vision Transformers (ViTs) have made significant progress in image classification, object detection, and many

---
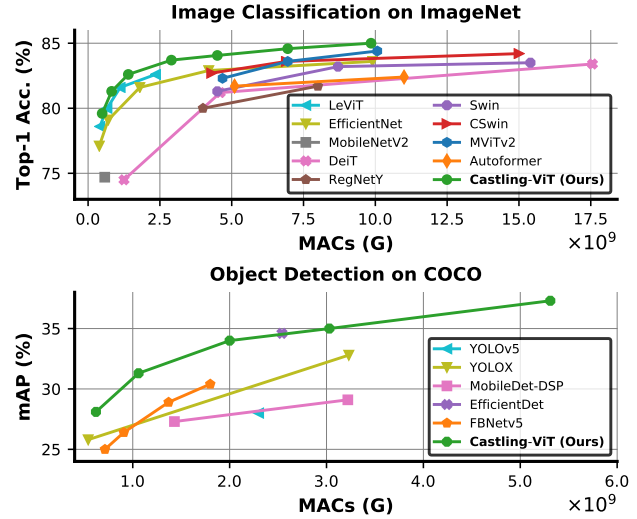*Equal contribution. † Work done while interning at Meta Research.

Figure 1. Castling-ViT over SOTA baselines on (1) ImageNet [18] image classification and (2) COCO [36] object detection.

other applications. It is well recognized that the superior performance achieved by ViTs is largely attributed to their self-attention modules that can better capture global context [20, 57, 64]. Nevertheless, ViTs' powerful self-attention module comes at the cost of quadratic complexity with the number of input tokens, causing a major efficiency bottleneck to ViTs' achievable runtime (i.e., inference latency) [3, 8, 32, 58, 66, 69, 76]. To mitigate this issue, linear attention designs have been developed to alleviate the vanilla ViT attention's quadratic complexity. In particular, existing efforts can be categorized into two clusters: (1) ViTs with local attention by restricting the attention window size [38, 53], sharing the attention queries [2], or representing the attention queries/keys with low rank matrices [58]; and (2) ViTs with kernel-based linear attention, which approximate the non-linearity softmax function by decomposing it into separate kernel embeddings. This enables a change in the matrix computation order for a reduced computational complexity [5, 6, 14, 29, 37, 39, 43, 66].

Despite their promise in alleviating ViTs' complexity

and thus inference runtime, both the local and linear attention compromise ViTs' performance due to the lack of capabilities to capture global or local context. To marry the best of both worlds, we advocate training ViTs with both (1) efficient but less powerful linear attention, i.e., without the high-order residuals in angular kernel expansion, and (2) powerful yet costly softmax-based masked attention. The latter helps approximate high-order residuals at the early training stage while being dropped during inference, based on an assumption that the remaining networks can gradually learn the high-order components at the later training stage [67]. This concept resembles the "castling" move in chess when two pieces are moved at once. While it sounds promising, there are still two challenges to achieve this. *First*, existing linear attention modules still underperform their vanilla softmax-based counterparts. Therefore, a better linear attention is crucial for the final performance. We find that angular kernels perform equally as softmax-based attentions in terms of similarity measurements. While they still suffer from a quadratic complexity, they can be divided into linear terms and high-order residuals. The challenge is *how to construct ViTs with only the linear terms. Second*, doing so would require that the trained ViTs merely rely on the linear terms towards the end of training, which would call for an approximation of the above high-order residuals. The challenge is that *how we can resort to costly but powerful modules to approximate high-order residuals during training but does not incur extra inference cost*.

In this work, we develop techniques to tackle those challenges, and make the following contributions:

- We propose a framework called **Castling-ViT**, which trains ViTs using both linear-angular attention and masked softmax-based quadratic attention, but then switches to having only linear-angular attentions during ViT inference to save computational costs.

- We develop a new linear-angular attention leveraging angular kernels to close the accuracy gap between linear attention and softmax-based attention. It expands angular kernels where linear terms are kept while complex high-order residuals are approximated.

- We use two parameterized modules to approximate the high-order residuals above: a depthwise convolution and an auxiliary masked softmax-based attention, where the latter's attention masks are regularized to gradually become zeros to avoid inference overhead.

- We conduct extensive experiments to validate the effectiveness of the proposed Castling-ViT. Results on classification, detection, and segmentation tasks consistently demonstrate its superior performance (↑1.8% top-1 accuracy or ↑1.2 mAP) or efficiency (40% MACs savings) over state-of-the-art (SOTA) CNNs and ViTs.

2. Related Works

Vision Transformers (ViTs). ViT [20] beats CNNs with a simple encoder-only transformer architecture taking the splitted non-overlapped image patches as sequential inputs, but relies on costly pretraining on a huge JFT-300M dataset [50]. Later, DeiT [54] and T2T-ViT [72] leverage an improved ViT training recipe or enhanced tokenization mechanism to achieve a comparable accuracy without the necessity of costly pretraining. To further improve ViTs' achievable accuracy-efficiency tradeoffs, CrossViT [9], PiT [26], PVT [59], MViT [21] and Swin-Transformer [38] propose a pyramid-like architecture, which is commonly used in CNNs [16, 28, 65]; DynamicViT [46], A-ViT [68], ToME [4], and MIA-Former [71] propose to adaptively identify and remove unnecessary input tokens for saving computational costs. With the goal of deploying ViTs in resource-constrained devices, various efficient ViT architectures have been proposed [6, 24, 35, 40]. For example, LeViT [24], CvT [63], and MobileViT [40] adopt more efficient self-attention implementation or incorporate convolutional feature extraction blocks into their early layers; EfficientFormer [35] further enables pure ViTs to run as fast as MobileNets. In contrast, our Castling-ViT explores whether ViTs can learn both global and local features while still being efficient at runtime. Also, we target a generic linear-angular attention that can serve as a drop-in replacement for all kinds of ViT architectures and thus is orthogonal to new ViT architecture designs.

Efficient ViT Variants. As commonly recognized, ViTs rely heavily on their self-attention module which is however costly due to its quadratic computational complexity with the total number of input tokens [29, 76]. To make the self-attention module more efficient, a surge of linear attention works have been proposed and can be roughly categorized into two groups: local attention [2, 38, 56, 58] or kernel-based linear attention [1, 5, 6, 14, 29, 37, 39, 43, 66]. For kernel-based linear attention, common designs approximate the softmax function [5, 14, 29] or the full self-attention matrix [39, 66] with orthogonal features or kernel embeddings, then the computation order can be changed from $(\mathbf{QK})\mathbf{V}$ to $\mathbf{Q}(\mathbf{KV})$. For example, [29] and [6] decompose the exponential terms in softmax-based attention into kernel functions and exchange the computation order. Despite their decent performance, currently kernel-based linear attention in general underperform the softmax-based attention. Recent works [8, 17] also unify low rank approximated and sparse attention (can also be dropped at inference) to improve ViTs' accuracy-efficiency tradeoffs. Different from the above works, we explore from a new perspective by taking spectral angles into consideration when measuring the similarities among tokens, resulting in linear-angular attention that can achieve comparable or even better performance than softmax-based attention. More efficient ViT variants

are supplied to the Appendix.

ViTs have also been used as the backbones for downstream tasks, e.g., detection [34, 60, 77] and segmentation [11, 42]. We supply more literature review to Appendix.

3. The Proposed Methods

3.1. Preliminary of Self-Attention

Self-Attention. Self-attention module is a core component of the Transformer [20, 57], and usually consists of multiple heads. Each head captures global-context information by measuring pairwise correlations among all N tokens (N denotes the total number of tokens) as defined below:

$$\mathbf{H}_t^k = \sum_{i=1}^{N} \frac{\exp(\mathbf{Q}_t \cdot \mathbf{K}_i^T / \sqrt{d})}{\sum_{j=1}^{N} \exp(\mathbf{Q}_t \cdot \mathbf{K}_j^T / \sqrt{d})} \cdot \mathbf{V}_i, \quad (1)$$

where $t \in \{1, \cdots, N\}, k \in \{1, \cdots, M\}, \mathbf{H}_t^k$ refers to the t-th row of the m-th head's attention matrix H. $\mathbf{Q}_t, \mathbf{K}_t, \mathbf{V}_t \in \mathbb{R}^d$, are the query, key, and value vectors obtained by linearly projecting the input $\mathbf{X_t} \in \mathbb{R}^D$ with three learnable weight matrices $W^Q, W^K, W^V \in \mathbb{R}^{D \times d}$. The attention head first computes the inner product between the query-key pairs, then scales the product results to stabilize the training and uses Softmax to normalize the resulting attention scores, and finally computes a weighted sum over all value vectors. Outputs from all attention heads are then concatenated together before a final linear projection with learnable weights. Note that $\exp(\cdot)$ denotes an exponential function. Computing Eq. (1) has a quadratic complexity of $\mathcal{O}(N^2)$.

Kernel-based Linear Attention. The core idea of linear attention [29, 49, 58] is to decompose the similarity measurement function into separate kernel embeddings, i.e., $\text{Sim}(\mathbf{Q}, \mathbf{K}) \approx \phi(\mathbf{Q})\phi(\mathbf{K})^T$, so that we can change the computation order to $\phi(\mathbf{Q})(\phi(\mathbf{K})^T \mathbf{V})$ based on the associative property of matrix multiplication. In this way, the attention complexity is quadratic to the feature dimension d instead of the token length N. One straightforward implementation of linear attention is to use Gaussian RBF kernels to measure the similarity, which can serve as an unbiased estimation of $\exp(\langle \cdot, \cdot \rangle)$ in Eq. (1):

$$\exp(\frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\sigma^2}) = \underbrace{\exp(\frac{-\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2})}_{\text{Gaussian RBF Kernel}} \exp(\frac{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2}{2\sigma^2}) \quad (2)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product operator. According to [45], we can induce a function $\phi(\cdot)$ to approximate the Gaussian RBF kernel, mapping the input space to the feature space. Assuming that both $\mathbf{Q}$ and $\mathbf{K}$ are normalized as unit row vectors along the feature dimension, then the attention formula can be approximated by:

$$\mathbf{H}_t^k = \frac{\phi(\mathbf{Q}_t) \cdot \sum_{i=1}^{N}(\phi(\mathbf{K}_i)^T \cdot \mathbf{V}_i)}{\phi(\mathbf{Q}_t) \cdot \sum_{j=1}^{N} \phi(\mathbf{K}_j^T)} \quad (3)$$
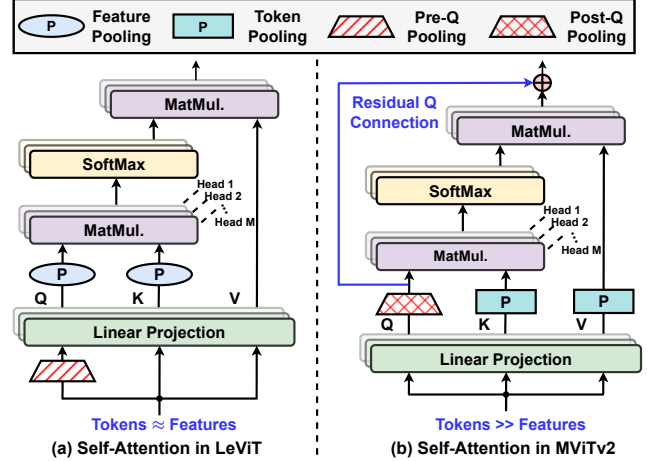

Figure 2. Illustration of self-attention used in LeViT and MViTv2.

While doing so can alleviate the attention complexity to become linear w.r.t. the token length N, it often causes a nontrivial accuracy drop as compared to the corresponding ViTs with vanilla softmax-based attention [6, 13, 72].

Revisit Attention Designs. Recent ViTs have achieved a low complexity for classification tasks (e.g., LeViT [24]) but still be costly for downstream tasks due to high input resolutions, i.e., the number of tokens (e.g., MViTv2 [34]). Therefore, one dilemma is the tradeoff between the model efficiency and generalizability. As illustrated in Fig. 2, LeViT shrinks the feature dimension to be more efficient on low-resolution tasks but cannot be well generalized to downstream tasks, due to the resulting (1) feature bottlenecks, i.e., insufficient feature dimensions, and (2) fixed attention biases. On the other hand, MViTv2 performs token pooling for better fitting high-resolution tasks but is less effective for low-resolution due to the caused token bottlenecks, i.e., insufficient number of tokens. Our ablation studies in Table 1 show that (1) for ImageNet classification, both token and feature pooling can lead to accuracy drops in ViTs; and (2) for downsampling layers, post-$\mathbf{Q}$ pooling (i.e., pooling after linear projection) together with residual connections in MViTv2 performs better than pre-$\mathbf{Q}$ pooling (i.e., pooling during linear projection) in LeViT. Therefore, we develop Castling-LeViTs on top of attention with merely post-$\mathbf{Q}$ pooling and residual connections.

3.2. The Proposed Castling-ViT Framework

Castling-ViT Overview. Fig. 4 illustrates an overview of the proposed Castling-ViT, which makes linear attention more powerful than previous designs while still being efficient during inference. In particular, we propose (1) a novel kernel-based linear-angular attention from the spectral angle perspective to close the accuracy gap between linear attention and softmax-based attention; and (2) a training augmentation method that leverages softmax-based attention as an auxiliary branch to assist the linear-angular atten-

Table 1. Analyze the attention design on ImageNet classification.

Pooling				Residual Q	MACs (M)	Top-1 Accuracy (%)
Token	Feat.	Pre-Q	Post-Q			
✓		✓			685	78.00
	✓	✓			661	78.19
		✓			736	79.11
✓			✓	✓	771	78.73
	✓		✓	✓	763	77.23
			✓	✓	838	80.05

tion only during ViT training. Note that a mask is applied to the auxiliary branch to manifest linear-angular attention and drop the auxiliary branch.

3.2.1 Linear-Angular Attention

Angular Kernel. In addition to the previously adopted polynomial, exponential, or RBF kernel that focuses on spatial similarity measurements [29, 37], we propose to consider measuring spectral similarity via angular kernel as an alternative to existing softmax-based attentions, leading to similar or better performance since it additionally takes into consideration the nature of spectral characteristics, e.g., the spectral angle as a distance measurement function [27, 30]. Such a spectral angle between two vectors is defined as:

$$\theta(\mathbf{x}_i, \mathbf{x}_j) = \arccos\left(\frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|}\right), \quad (4)$$

where $\|\cdot\|$ is the Euclidean distance and $\langle\cdot,\cdot\rangle$ is the inner product. The output range of θ is $[0, \pi]$. Such an angle can be used as a distance. In our design, we define the angular kernel as a similarity measurement function between the queries $\mathbf{Q}$ and keys $\mathbf{K}$ as:

$$\text{Sim}(\mathbf{Q}_i, \mathbf{K}_j) = 1 - \frac{1}{\pi} \cdot \theta(\mathbf{Q}_i, \mathbf{K}_j), \quad (5)$$

and the output range is thus $[0, 1]$. With more aligned $\mathbf{Q}_i$ and $\mathbf{K}_j$, θ is closer to 0 and thus the similarity is closer to 1; In contrast, if $\mathbf{Q}_i$ and $\mathbf{K}_j$ have opposite features, θ is closer to π and thus the similarity is closer to 0.

Properties of Angular Kernel and Its Feature Space. One property of our angular kernel is that replacing the similarity measurement in self-attention with such a kernel provides an efficient way to implicitly map the input data to a high (even infinite after expansion) dimensional feature space [27], where the distances/angles are calculated based on a rich feature structure. Let $\phi(\cdot)$ denotes the implicit map induced by this kernel, the norm of mapped input data is:

$$\|\phi(\mathbf{x}_i)\|^2 = \text{Sim}(\mathbf{x}_i, \mathbf{x}_i) = 1 - \frac{1}{\pi} \cdot 0 = 1, \quad (6)$$

which means that all data in the input space are mapped onto the sphere of radius 1 in the feature space. Also, the distance between two input features is given by:

$$\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 = 1 + 1 - 2 \cdot \phi(\mathbf{x}_i)\phi(\mathbf{x}_j)^T$$
$$= 2 \cdot (1 - \text{Sim}(\mathbf{x}_i, \mathbf{x}_j)) = \frac{2}{\pi} \cdot \theta(\mathbf{x}_i, \mathbf{x}_j). \quad (7)$$

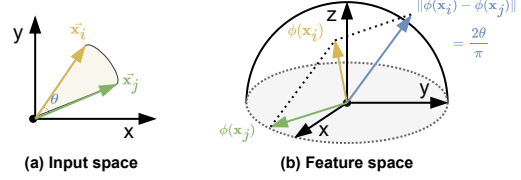**(a) Input space**　　**(b) Feature space**

Figure 3. (a) A simple 2D input space and (b) the corresponding 3D feature space after applying the angular kernel to the 2D input data. Note that real input data/features are of higher dimensions.

That is, the square (Euclid) distance and the spectral angle is positively correlated and the distance range is $[0, 2]$. As shown in Fig. 3, the angles in the input space is correlated to the feature distance after applying our angular kernel.

Expansion of Angular Kernel. A natural following question is how to incorporate the above angular kernel for designing linear attention given its quadratic complexity w.r.t. the input token length. Recalling from trigonometric identities and the expansion of the arccos function into an infinite series, we reformulate the similarity function as:

$$\text{Sim}(\mathbf{Q}_i, \mathbf{K}_j) = 1 - \frac{1}{\pi} \cdot \left(\frac{\pi}{2} - \arcsin\left(\mathbf{Q}_i \cdot \mathbf{K}_j^T\right)\right)$$
$$= \underbrace{\frac{1}{2} + \frac{1}{\pi} \cdot \left(\mathbf{Q}_i \cdot \mathbf{K}_j^T\right)}_{\text{Linear-angular terms}}$$
$$+ \underbrace{\frac{1}{\pi} \cdot \sum_{k=1}^{\infty} \frac{(2k)!}{2^{2k}(k!)^2(2k+1)} \left(\mathbf{Q}_i \cdot \mathbf{K}_j^T\right)^{2k+1}}_{\text{High-order residuals when } k \geq 1 \text{ (non-linear)}}, \quad (8)$$

where $\mathbf{Q}_i \cdot \mathbf{K}_j^T$ denotes the normalized linear kernel function $\langle \mathbf{Q}_i, \mathbf{K}_j \rangle / \|\mathbf{Q}_i\| \cdot \|\mathbf{K}_j\|$, which is equivalent to the inner product if $\mathbf{Q}_i$ and $\mathbf{K}_j$ are unit vectors. We see that the first linear-angular terms can be directly used as the similarity measurement in linear attention, while the remaining higher-order terms of an infinite series introduce a much higher complexity. As such, we propose to adopt a relaxation to approximate it.

Linear-Angular Attention. To construction our linear-angular attention, we leverage parametrized DNN modules to approximate the expectation of the high-order residuals, i.e., $\mathbf{M}_{ij} = \mathbb{E}_{\mathbf{Q}_i \sim p_{Q_i}, \mathbf{K}_j \sim p_{K_j}} \left(\sum_{k=1}^{\infty} \alpha_k (\mathbf{Q}_i \cdot \mathbf{K}_j^T)^{2k+1}\right)$, where α_k is the coefficient for the k-th order term in Eq. (8). The key question is *how to design such DNN modules*. Kim et al. [31] conducted an analysis on average attention weights of ViT models, which show a strong inductive bias to attend to neighboring tokens. To capture this, we introduce a learnable depthwise convolution (DWConv) module which is applied on all the value tokens, so $\mathbf{M}_{ij}$ by-design attends to nearby tokens, as illustrated in Fig. 4. Second, DWConv is limited by its receptive field [41] while the averaged attention map also has a small number of off-diagonal scores to connect to nonadjacent tokens [31]. To capture that, we adopt a sparse softmax-based attention as
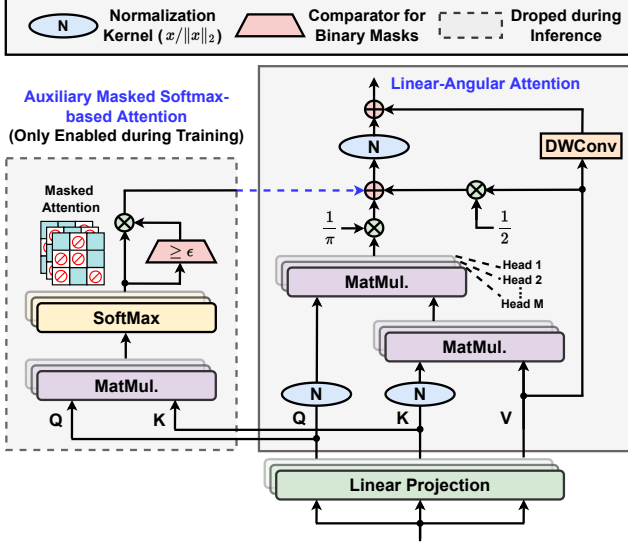
Figure 4. The visualization of our Castling-ViT with both linear-angular attention and auxiliary softmax-based self-attention.

described in Section 3.2.2. In practice, we found that such a sparse softmax attention tends to converge to all zeros after adding a simple threshold regularization. We provide more explanation and visualization in Sec. 3.2.2 and Sec. 4.5.

In this way, our attention module can be formulated as:

$$
\begin{aligned}
\mathbf{H} = \text{Sim}(\mathbf{Q}, \mathbf{K}) \cdot \mathbf{V} &\approx \frac{1}{2} \cdot \mathbf{V} + \frac{1}{\pi} \cdot \mathbf{Q} \cdot (\mathbf{K}^T \cdot \mathbf{V}) \\
&+ \mathbf{M}_{DW} \cdot \mathbf{V} + \mathbf{M}_{SparseAttn},
\end{aligned} \quad (9)
$$

where $1/\pi \cdot Q \cdot (K^T \cdot V)$ is the linear term with $\mathcal{O}(N)$ complexity, $\mathbf{M}_{DW} \cdot V$ is the matrix form of DWConv, and $\mathbf{M}_{SparseAttn}$ is the normalized sparse softmax attention. The overall complexity to compute Eq. (9) is linear to the input token length, where the MACs of DWConv is also negligible (e.g., $< 1\%$ of the total MACs). Also, normalizations ($x/\|x\|_2$) are inserted to the $\mathbf{Q}/\mathbf{K}$ and sparse attention branches to help the similarity measurement following [5].

3.2.2 Switch Towards Linear-Angular Attention

Recall that we add a sparse softmax-based attention as an auxiliary branch in the linear-angular attention to help approximate the high-order residuals. Such a costly attention can be potentially dropped without hurting the inference accuracy, drawing inspiration from [67] that the remaining network can gradually learn the high-order/frequency components at the later training stage. Next, we explain how it is constructed and dropped.

Sparse Training Augmentation. As illustrated in Fig. 4, we adopt a masked softmax-based attention as an auxiliary branch to augment ViT training. In particular, we first use a comparator with a predefined threshold ϵ to generate a binary mask, where attentions greater than ϵ are set to 1 and 0 otherwise. These masks are then applied to the attention maps to generate masked attention maps that will be

summed up with our linear-angular attention together as the final attention. The resulting sparse attention is given by:

$$
\mathbf{M}_{SparseAttn}(\mathbf{Q}, \mathbf{K}) = \text{Mask}_\epsilon(\text{Softmax}(\mathbf{Q} \cdot \mathbf{K}^T)). \quad (10)
$$

where $\text{Mask}_\epsilon(x) = x$ if $x > \epsilon$ else 0, acting as an element-wise threshold function. As such, the sparse attention captures the higher attention scores (i.e., strong local features), and can potentially complement our linear-angular attention by supplementing the missing higher-order terms. Such an assumption aligns well with recent findings that (1) low-rank and sparse approximations complement each other [8, 17, 31]; and (2) linear attention lacks local feature extraction capabilities over its softmax-based counterpart [6].

Castling During ViT Inference. As we are targeting efficient ViT inference, it is desired to reduce or completely remove the costly softmax-based attentions while only keeping our linear-angular attention at runtime, i.e., performing castling. In our experiments, we found that under the sparsity regularization above, the softmax attention naturally converges to all zeros as the training progresses. We consider both fixed and dynamic schedules for ϵ and find that our Castling-ViT is not sensitive to neither the threshold value nor the threshold schedule for a given task. Given a fixed mask threshold (e.g., $\epsilon = 0.02$ in image classification experiments), the masks become all zeros at latter training stages and thus the auxiliary branch can be removed without hurting the model accuracy. We supply the visualization of the mask-evolving trajectory and our conjecture for understanding such a phenomenon in Sec. 4.5.

4. Experiments

4.1. Experiment Settings

Tasks, Datasets, and Models. *Tasks and Datasets.* We consider three benchmark datasets and three representative vision tasks to demonstrate the superiority of the proposed Castling-ViT, including image classification on ImageNet dataset [18] with 1.2 million training and 50K validation images; Object detection on COCO dataset [36] with 118K training and 5K validation images; Semantic segmentation on ADE20K dataset [75] with 20K/2K/3K images for training, validation, and testing, respectively. *Models.* We apply our proposed Castling-ViT idea on top of various models. For the classification task, we consider LeViT [24], MViTv2 [34], and DeiT [54]; For the detection task, we consider models with efficient ViT backbones (e.g., PicoDet [70] with modified ESNet and LCNet backbones [15] with transformer blocks); For the segmentation task, we consider Mask2former [11] with ViT-Base backbone.

Training Settings. *For the classification task,* we use a SGD optimizer with 0.9 momentum and 2×10^{-5} weight decay to train ViTs for 1000 epochs using 64 V100 GPUs, with each card having 64 (LeViT) or 32 (MViTv2/DeiT) batch sizes. The learning rate is 2.0

Table 2. Castling-ViT over SOTA baselines on ImageNet.

MACs Ranges	Models	Params (M)	MACs (G)	Top-1 Acc.(%)	Top-5 Acc.(%)
<1G	MobileNetV2-1.4 [47]	6.9	0.58	74.7	-
	EfficientNet-B0 [51]	5.3	0.39	77.1	93.3
	EfficientNet-B1 [51]	7.8	0.70	79.1	94.4
	MobileViT-XS [40]	2.3	0.70	74.8	92.3
	LeViT-128 [24]	9.2	0.41	78.6	-
	LeViT-192 [24]	10.9	0.66	80.0	-
	Castling-LeViT-128	10.5	0.49	**79.6**	**94.6**
	Castling-LeViT-192	12.7	0.82	**81.3**	**95.5**
1~3G	EfficientNet-B3 [51]	12.0	1.80	81.6	95.7
	HRFormer-T [73]	8.0	1.80	78.5	-
	DeiT-T [54]	5.6	1.25	74.5	91.9
	LeViT-256 [24]	18.9	1.12	81.6	-
	LeViT-384 [24]	39.1	2.35	82.6	-
	Caslting-DeiT-T	5.64	1.18	76.0	92.5
	Castling-LeViT-256	22.0	1.40	**82.6**	**96.1**
	Castling-LeViT-384	45.8	2.90	**83.7**	**96.7**
3~10G	RegNetY-4G [44]	21.0	4.00	80.0	-
	RegNetY-8G [44]	39.0	8.00	81.7	-
	EfficientNet-B4 [51]	19.0	4.20	82.9	96.4
	EfficientNet-B5 [51]	30.0	9.90	83.6	96.7
	Swin-T [38]	29.0	4.50	81.3	95.5
	Swin-S [38]	50.0	8.70	83.2	96.2
	CSWin-T [19]	23.0	4.30	82.7	-
	CSWin-S [19]	35.0	6.90	83.6	-
	DeiT-S [54]	21.9	4.60	81.2	95.4
	AutoFormer-S [10]	22.9	5.10	81.7	95.7
	PVTv2-V2 [61]	25.0	4.00	82.0	-
	MViTv2-T [34]	24.0	4.68	82.3	-
	MViTv2-S [34]	34.7	6.95	83.6	-
	Castling-MViTv2-T	24.1	4.50	**84.1**	**96.8**
	Castling-MViTv2-S	34.7	6.95	**84.6**	**97.0**
>10G	CaiT-S36 [55]	68.0	13.90	83.3	-
	Swin-B [38]	88.0	15.40	83.5	96.5
	CSWin-B [19]	78.0	15.00	84.2	-
	AutoFormer-B [10]	54.0	11.00	82.4	95.7
	MViTv2-B [34]	51.2	10.07	84.4	-
	DeiT-B [54]	86.3	17.56	83.4	96.5
	Castling-DeiT-B	87.22	17.28	**84.2**	-
	Castling-MViTv2-B	51.9	9.82	**85.0**	**97.2**

Figure 5. Castling vs. baseline ViT on ImageNet.

for the classification task; AP, AP^{50}, AP^{75} for the detection task (AP: average precision); mIoU, mAcc, and pAcc for the segmentation task (mIoU: mean intersection over union). For efficiency metrics, we compare the number of model parameters or inference FLOPs (or MACs).

4.2. Castling-ViT over SOTA Baslines

Image Classification. To evaluate the effectiveness of our proposed techniques on the image classification task, we apply the proposed Castling-ViT idea to three typical or SOTA ViT architectures: DeiT [54] as typical ViTs, LeViT [24] as representative efficient ViTs, and MViTv2 [34] as representative hierarchical ViTs for downstream applications of high input resolutions, and compare their performance over baselines on ImageNet. As shown in Tab. 2, the comparison across a large MACs (or FLOPs) range starting from 0.4G to 17G. We categorize ViT models into four regimes: <1G, 1~3G, 3~10G, and >10G, and select baselines in each regime to benchmark separately for the clarity purpose. Castling-ViTs consistently perform better than all baselines across various MACs ranges in terms of the accuracy-efficiency tradeoff. For example, Castling-LeViT achieves 82.6% top-1 accuracy with only 1.40G MACs while LeViT requires 2.35G FLOPs instead, i.e., ↓**40% MACs**; On the other hand, under comparable MACs, Castling-MViTv2 achieves 84.1% accuracy vs. MViTv2 with 82.3% accuracy instead, i.e., ↑**1.8% top-1 accuracy**. Overall, Castling-ViT achieves an improved accuracy of ↑**0.5%** ~ ↑**6.6%**, ↑**1.0%** ~ ↑**8.1%**, ↑**1.0%** ~ ↑**4.1%**, and ↑**0.6%** ~ ↑**2.6%** over baselines under<1G MACs, 1~3G MACs, 3~10G MACs, and >10G MACs, respectively. Note that we calculate improvements under comparable MACs. In addition to the overall comparison, we also visualize the apple-to-apple benchmark, e.g., Castling-LeViT vs. LeViT, to validate the effectiveness of proposed techniques. As shown in Fig. 5, Castling-LeViT/MViTv2 achieves ↓**25.7%** ~ ↓**55.3%** MACs reductions under comparable accuracies or offers a comparable or better accuracy (↑**0.6%** ~ ↑**1.8%**) under comparable MACs over corresponding LeViT/MViTv2 baselines.

Object Detection. We also extend the Castling-ViT to the downstream object detection task and compare it with previous efficient detectors on COCO dataset to evaluate its efficacy. Specifically, we construct the detector with modified ESNet [70] or LCNet [15] (replace the last one or two

with first 11 epochs warm-up starting from 0.01 and decays by a factor of 0.9875 per epoch [62]. Also, we use the distillation [54] based on a teacher model with a 85.5% accuracy. *For the detection task,* we adopt SGD optimizer with momentum 0.9 and weight decay 4e-5 to train models on COCO. All models are trained on 8 V100 GPUs with each card having 80 batch sizes following PicoDet's training recipe [70]. Also, we follow LeViT's training recipe [24] to pretrain backbones on ImageNet; *For the segmentation task,* we follow Mask2former's training recipe [11] to train models on ADE20K, where ViT backbones are pretrained following MAE [25] if specified.

Baselines and Evaluation Metrics. *Baselines.* For the classification task, we compare the proposed Castling-ViT with LeViT [24], MviTv2 [34], DeiT [54], Swin [38], CSWin [19], PVT [59], etc. For the detection task, we compare with FBNetV5 [62], YOLOX [23], YOLOv5, MobileDet [65], and EfficientDet [52]. For the segmentation task, we compare with Mask2former [11] with ViT backbones. *Evaluation Metrics.* We evaluate the Castling-ViT and all baselines in terms of accuracy-efficiency tradeoffs. Specifically, the accuracy metrics refer to top-1/5 accuracy
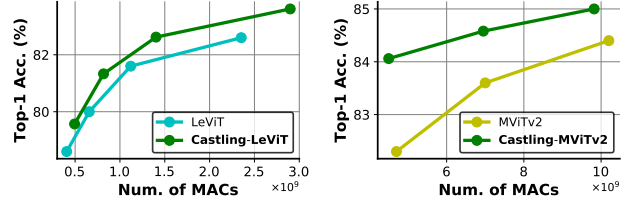
Table 3. Castling-ViT over SOTA baselines on COCO detection, where Castling-ViT[†] means that the detection model adopts modified ESNet (use transformer blocks in the last stage; referred to ESNet-ViT) as backbone while Castling-ViT[⋆] uses modified LCNet (use transformer blocks in the last two stage; referred to LCNet-ViT) as backbone. Also, "-S/M/L" and "-320/416" represent small/medium/large model variants and 320x320/416x416 input resolutions, respectively.

Models	Params (M)	MACs (G)	mAP	AP50	AP75
MobileDet-DSP [65]	4.85	1.43	27.3	-	-
MobileDet-DSP-Fused [65]	9.15	3.22	29.1	-	-
EfficientDet-D0 [52]	3.90	2.54	34.6	53.0	37.1
YOLOv5-N	1.90	2.30	28.0	45.7	-
YOLOX-Nano [23]	0.91	0.54	25.8	-	-
YOLOX-Tiny [23]	5.06	3.23	32.8	-	-
FBNetV5-AC-224x320 [62]	-	0.71	25.0	-	-
FBNetV5-AR-224x320 [62]	-	0.91	26.4	-	-
FBNetV5-A-224x320 [62]	-	1.35	27.2	-	-
FBNetV5-AC-320x640 [62]	-	1.37	28.9	-	-
FBNetV5-AR-320x640 [62]	-	1.80	30.4	-	-
Castling-ViT-S-320[†]	**3.25**	**0.62**	**28.1**	**42.3**	**29.2**
Castling-ViT-S-416[†]	**3.25**	**1.06**	**31.3**	**46.7**	**32.5**
Castling-ViT-M-416[†]	**6.01**	**2.00**	**34.0**	**49.5**	**35.9**
Castling-ViT-L-416[†]	**9.32**	**3.03**	**35.0**	**50.7**	**37.2**
Castling-ViT-L-416[⋆]	**13.10**	**5.31**	**37.3**	**53.4**	**39.6**

Table 4. Castling-ViT over baselines on the ADE20K segmentation dataset, serving as backbones in the Mask2former framework. Note that we report both total MACs and backbone MACs, i.e., (·).

Mask2former w/	MAE Pretrain	Params (M)	MACs (G)	mIoU	mAcc	pAcc
ViT-Base	N	118	229 (182)	34.54	46.36	75.84
Castling-ViT-Base	N	118	**195 (147)**	**34.67**	**46.47**	**76.20**
ViT-Base	Y	118	229 (182)	47.92	61.00	83.02
Castling-ViT-Base	Y	118	**195 (147)**	**48.44**	**61.82**	**83.29**

Table 5. Ablation studies of Castling-ViT with various kernel-based linear attentions, where MC refers to mean-centering kernel function applied to $\mathbf{Q}/\mathbf{K}$; Softmax also means the kernel function; ReLU-S and ReLU-E denote applying ReLU as kernels for decomposing softmax or exponential terms, respectively. All models are trained on COCO from scratch without pretraining.

Backbone	Linear Attn	S-320		S-416		M-416	
		mAP	AP50	mAP	AP50	mAP	AP50
LCNet	N/A	25.5	39.0	29.7	44.2	34.3	49.9
LCNet-ViT	N/A	25.9	39.6	29.8	44.4	34.9	50.4
LCNet-ViT	MC [17]	22.8	34.9	26.3	39.6	31.5	46.1
	Softmax [49]	26.0	39.6	29.2	43.4	34.4	49.8
	Cosine [5]	24.4	37.4	27.2	41.0	32.9	48.0
	ReLU-S	23.9	36.7	-	-	30.9	45.3
	ReLU-E [6]	26.5	40.2	30.2	44.7	34.3	49.7
	Lin.-Angular	**27.0**	**40.6**	**30.4**	**45.5**	**35.5**	**51.1**

stages with transformer blocks) as backbones and follow PicoDet [70]'s detection head design as well as their training recipe. Fig. 1 and Tab. 3 show the overall comparison between the proposed Castling-ViT and other baselines. We can see that our Castling-ViT consistently achieves better accuracy-efficiency tradeoffs, leading to ↑**6.0**, ↑**2.2** ∼ ↑**2.3**, ↑**4.0** ∼ ↑**5.9**, ↑**3.1** ∼ ↑**4.0 mAP** improvements as compared to YOLOv5, YOLOX [23], MobileDet [65], and FBNetv5 [62], respectively, under comparable or even less MACs. As for EfficientDet [52], our method achieves comparable accuracy-efficiency trade-offs. Apart from the overall comparison with baselines, we also provide the apple-to-apple comparison with detectors with softmax-based attention, we supply the comparison results and Castling-ViT's breakdown analysis to Sec. 4.4. This set of experiments validate the effectiveness of the proposed Castling-ViT for serving as efficient detector backbones (e.g., ≤3G MACs) in the object detection task.

Semantic Segmentation. We further extend Castling-ViT to the semantic segmentation task to evaluate its effectiveness. Specifically, we use ViT-Base as the backbone in Mask2former [11] framework to serve as our baseline and testbed. Then we build the Castling-ViT-Base as the backbone to benchmark on the ADE20K dataset. As shown in Tab. 4, Mask2former with Castling-ViT backbone achieves **15%** total MACs reductions and **19%** backbone MACs reduction under comparable or slightly better mIoU, i.e., ↑**0.13%** and ↑**0.52%** without or with MAE pretraining on ImageNet [25]. This set of experiments validate that our proposed techniques could be well generalized to various downstream tasks that require large input resolutions.

4.3. Linear-Angular Attention over SOTA Baselines

We also conduct ablation studies among various kinds of kernels used in linear attention to evaluate the superiority of our proposed linear-angular kernel. Also shown in Tab. 5, we compare the angular kernel with five other commonly adopted kernels, results on three model and resolution settings consistently demonstrate that our proposed angular kernel helps achieve better mAP, e.g., ↑4.0% ∼ ↑4.2% over MC [17], ↑1.0% ∼ ↑1.2% over Softmax [49], ↑2.6% ∼ ↑3.2% over Cosine [5], ↑3.1% ∼ ↑4.6% and ↑0.2% ∼ ↑1.2% over ReLU-S and ReLU-E [6], respectively. This set of experiments validate the superiority of the proposed linear-angular kernels.

4.4. Ablation Studies of Castling-ViT

We conduct ablation studies on Castling-ViT's linear-angular attention, added DWConv, as well as auxiliary masked softmax-based attention, as shown in Tab. 6 and Tab. 7, where the experiments are performed on Castling-ViT-S-320[⋆]/M-416[⋆], respectively, without pretraining on ImageNet. Note that here [⋆] means that we use LCNet-ViT, i.e., replacing all layers in last two stages with transformer blocks, as backbones. Results in three tables show detailed performance breakdown and consistently demonstrate that all components in our proposed Castling-ViT contribute to the final performance. Specifically, linear-angular attention itself already achieves ↓0.1% ∼ ↑0.5% mAP improvements while leading to 10.1% ∼ 15.3% MACs reductions simultaneously. Adding DWConv in the middle of MLP layers further leads to ↓0.1% ∼ ↑0.6% mAP improvements while only incur negligible MACs overhead. Adding aux-

Table 6. Ablation studies of Castling-ViT-S-320 w/ LCNet backbone, or LCNet-ViT backbone + linear-angular attention (denoted as Lin.) + DWConv (denoted as DW.) + auxiliary softmax-based attention (denoted as SparseAttn.).

Castling-ViT-S-320 w/	Params (M)	MACs (G)	mAP	AP50	AP75
LCNet	1.10	0.48	25.5	39.0	26.5
LCNet-ViT	2.14	0.69	25.9	39.6	26.9
+ Lin.	2.14	0.62	26.2	39.7	27.3
+ Lin.+DW.	2.14	0.62	26.8	40.7	28.0
+ Lin.+DW.+SparseAttn.	2.14	0.62	27.0	40.6	28.0

Table 7. Ablation studies of Castling-ViT-M-416 w/ LCNet backbone, or LCNet-ViT backbone + Lin. + DW. + SparseAttn.

Castling-ViT-M-416 w/	Params (M)	MACs (G)	mAP	AP50	AP75
LCNet	3.32	2.15	34.3	49.9	36.5
LCNet-ViT	7.53	3.51	34.9	50.4	37.0
+ Lin.	7.53	3.15	34.8	50.3	36.7
+ Lin.+DW.	7.53	3.15	35.1	50.9	36.9
+ Lin.+DW.+SparseAttn.	7.53	3.15	35.3	51.1	37.3

iliary masked softmax-based attention further increases the mAP by 0.2% without incurring any overhead since it only assists the training while being removed during inference. Note that although LCNet-ViT backbone leads to better mAP, it also introduces more MACs as compared to efficient convolutions. As such, LCNet-ViT still slightly underperforms LCNet backbones in terms of accuracy-efficiency trade-offs, since we directly replace the CNN backbones with transformer blocks, whose architecture could not be optimal for ViTs. It remains to an open problem to build ViT based models that achieves higher accuracy-efficiency trade-off than pure ConvNet models.

4.5. Discussion on the Auxiliary Branch

Trajectory of Nonzeros in Masks. To further understand the effect of the auxiliary branch, we visualize the trajectories of nonzeros in masks of auxiliary softmax-based attention. As shown in Fig. 6, we count the nonzeros in the first attention layer's masks throughout the training of both Castling-ViT-S-416

Figure 6. Visualizing the trajectories of nonzeros in auxiliary masks during training.

and Castling-ViT-M-416 with the LCNet-ViT backbone on COCO. we observe that the introduced auxiliary attention will only assist the training in the early or middle training stages and will gradually vanish towards all zeros in the later training stage, which is well aligned with the assumption in our Castling-ViT. This set of experiments validate the idea of performing "castling" in ViTs, i.e., drop the auxiliary branch without sacrificing performance.
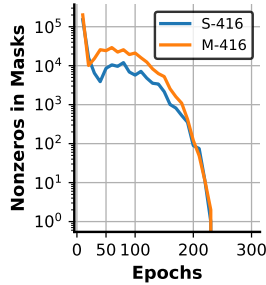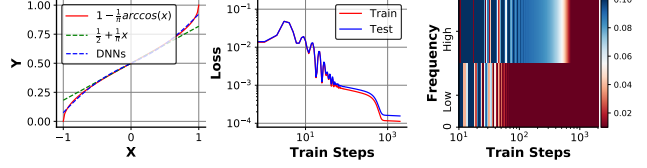

Figure 7. **Left:** Visualizing the taget similarity function, linear-angular terms, and DNNs' fitting curve; **Middle:** Visualizing the loss trajectory during DNN training; **Right:** Visualizing the learning trajectory of both low and high frequency components.

Conjecture of Castling. We conduct a synthetic experiment to give an analogy for explaining the castling phenomenon. In Fig. 7 (Left), we visualize the curve of (1) angular similarity function (denoted in Red), (2) linear-angular terms (denoted in Green), and (3) two-layer DNN's approximation. We see that only keeping linear-angular terms leads to distortion, while DNN is capable of learning the missing high-frequency parts, whose loss trajectory is shown in Fig. 7 (Middle). Also, the learning trajectory of frequency components is visualized in Fig. 7 (Right), following F-Principle [67], where x-axis is training steps, y-axis categorizes both low and high frequency components, blue/red colors refer to large/small difference between the learned frequency components and target frequency components, respectively. We observe that DNNs fit target functions from low to high frequencies during training. It indicates that DWConv itself is not sufficient for approximating the high-order residual at early training stages. Costly attention that contains high-order residuals is then desired to help training while being dropped at inference since the remaining networks can gradually learn the high-frequency components at later training stages [67].

5. Conclusion

We present Castling-ViT that trains ViTs with both linear-angular and softmax-based quadratic attention but switches to only having the former during inference. Castling-ViT leverages angular kernels to measure the similarities between the queries and keys via spectral angles and highlights two enablers: (1) a new linear-angular attention mechanism: we decompose angular kernels to linear terms and high-order residuals, and keep only the former for inference; and (2) we approximate the high-order residuals using a depthwise convolution and an auxiliary masked softmax attention whose masks gradually become zeros during training without incurring inference overhead. Extensive experiments consistently validate Castling-ViT's advantages.

Acknowledgment

References

[1] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. Advances in neural information processing systems, 34:20014–20027, 2021. 2, 13

[2] Moab Arar, Ariel Shamir, and Amit H Bermano. Learned queries for efficient local attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10841–10852, 2022. 1, 2, 13

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150, 2020. 1, 13

[4] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. arXiv preprint arXiv:2210.09461, 2022. 2

[5] 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. 1, 2, 5, 7, 13

[6] Han Cai, Chuang Gan, and Song Han. Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition. arXiv preprint arXiv:2205.14756, 2022. 1, 2, 3, 5, 7, 13, 14

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In European conference on computer vision, pages 213–229. Springer, 2020. 13

[8] Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention. Advances in Neural Information Processing Systems, 34:17413–17426, 2021. 1, 2, 5, 13

[9] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. arXiv preprint arXiv:2103.14899, 2021. 2

[10] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 12270–12280, 2021. 6

[11] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1290–1299, 2022. 3, 5, 6, 7, 13

[12] 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. 13

[13] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. arXiv preprint arXiv:2009.14794, 2020. 3

[14] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In International Conference on Learning Representations, 2021. 1, 2, 13

[15] Cheng Cui, Tingquan Gao, Shengyu Wei, Yuning Du, Ruoyu Guo, Shuilong Dong, Bin Lu, Ying Zhou, Xueying Lv, Qiwen Liu, et al. Pp-lcnet: A lightweight cpu convolutional neural network. arXiv preprint arXiv:2109.15099, 2021. 5, 6

[16] Xiaoliang Dai, Alvin Wan, P. Zhang, B. Wu, Zijian He, Zhen Wei, K. Chen, Yuandong Tian, Matthew E. Yu, Péter Vajda, and J. Gonzalez. Fbnetv3: Joint architecture-recipe search using neural acquisition function. ArXiv, abs/2006.02049, 2020. 2

[17] Jyotikrishna Dass, Shang Wu, Huihong Shi, Chaojian Li, Zhifan Ye, Zhongfeng Wang, and Yingyan Lin. Vitality: Unifying low-rank and sparse approximation for vision transformer acceleration with a linear taylor attention. In The 29th IEEE International Symposium on High-Performance Computer Architecture (HPCA 2023), 2023. 2, 5, 7, 14

[18] 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. 1, 5

[19] 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. 6

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain 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. 1, 2, 3

[21] 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

[22] Peng Gao, Teli Ma, Hongsheng Li, Jifeng Dai, and Yu Qiao. Convmae: Masked convolution meets masked autoencoders. arXiv preprint arXiv:2205.03892, 2022. 13

[23] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430, 2021. 6, 7, 14

[24] Ben 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. arXiv preprint arXiv:2104.01136, 2021. 2, 3, 5, 6, 13, 14

[25] 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. 6, 7, 13

[26] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. arXiv preprint arXiv:2103.16302, 2021. 2

[27] Paul Honeine and Cédric Richard. The angular kernel in machine learning for hyperspectral data classification. In 2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, pages 1–4. IEEE, 2010. 4

[28] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1314–1324, 2019. 2

[29] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In International Conference on Machine Learning, pages 5156–5165. PMLR, 2020. 1, 2, 3, 4, 13

[30] Nirmal Keshava. Distance metrics and band selection in hyperspectral processing with applications to material identification and spectral libraries. IEEE Transactions on Geoscience and remote sensing, 42(7):1552–1565, 2004. 4

[31] Kyungmin Kim, Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Zhicheng Yan, Peter Vajda, and Seon Joo Kim. Rethinking the self-attention in vision transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3071–3075, 2021. 4, 5

[32] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. arXiv preprint arXiv:2001.04451, 2020. 1, 13

[33] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. arXiv preprint arXiv:2203.16527, 2022. 13

[34] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4804–4814, 2022. 3, 5, 6, 13, 14

[35] 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. 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 European conference on computer vision, pages 740–755. Springer, 2014. 1, 5

[37] Jing Liu, Zizheng Pan, Haoyu He, Jianfei Cai, and Bohan Zhuang. Ecoformer: Energy-saving attention with linear complexity. In NeurIPS, 2022. 1, 2, 4, 13

[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, 6, 13, 14

[39] 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. 1, 2, 13

[40] Sachin Mehta and Mohammad Rastegari. Mobilevit: lightweight, general-purpose, and mobile-friendly vision transformer. arXiv preprint arXiv:2110.02178, 2021. 2, 6

[41] John Moody and Christian Darken. Learning with localized receptive fields. Yale Univ., Department of Computer Science, 1988. 4

[42] Zipeng Qin, Jianbo Liu, Xiaolin Zhang, Maoqing Tian, Aojun Zhou, Shuai Yi, and Hongsheng Li. Pyramid fusion transformer for semantic segmentation. arXiv preprint arXiv:2201.04019, 2022. 3, 13

[43] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. In International Conference on Learning Representations, 2022. 1, 2, 13

[44] 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

[45] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. Advances in neural information processing systems, 20, 2007. 3

[46] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. Advances in neural information processing systems, 34:13937–13949, 2021. 2

[47] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4510–4520, 2018. 6

[48] Robert R. Schaller. Moore's law: past, present and future. IEEE spectrum, 1997. 14

[49] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 3531–3539, 2021. 3, 7

[50] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In ICCV, 2017. 2

[51] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, pages 6105–6114. PMLR, 2019. 6

[52] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10781–10790, 2020. 6, 7

[53] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In International Conference on Learning Representations, 2022. 1

[54] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 10347–10357. PMLR, 18–24 Jul 2021. 2, 5, 6, 14

[55] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 32–42, 2021. 6

[56] 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. 2, 13

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. 1, 3

[58] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020. 1, 2, 3, 13

[59] 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. arXiv preprint arXiv:2102.12122, 2021. 2, 6, 14

[60] 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 (ICCV), pages 568–578, October 2021. 3, 13

[61] 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. 6

[62] Bichen Wu, Chaojian Li, Hang Zhang, Xiaoliang Dai, Peizhao Zhang, Matthew Yu, Jialiang Wang, Yingyan Lin, and Peter Vajda. Fbnetv5: Neural architecture search for multiple tasks in one run. arXiv preprint arXiv:2111.10007, 2021. 6, 7

[63] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. arXiv preprint arXiv:2103.15808, 2021. 2

[64] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. Lite transformer with long-short range attention. In International Conference on Learning Representations, 2020. 1

[65] Yunyang Xiong, Hanxiao Liu, Suyog Gupta, Berkin Akin, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Vikas Singh, and Bo Chen. Mobiledets: Searching for object detection architectures for mobile accelerators. arXiv preprint arXiv:2004.14525, 2020. 2, 6, 7

[66] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 14138–14148, 2021. 1, 2, 13

[67] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. arXiv preprint arXiv:1901.06523, 2019. 2, 5, 8

[68] Hongxu Yin, Arash Vahdat, Jose M 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, pages 10809–10818, 2022. 2

[69] Haoran You, Zhanyi Sun, Huihong Shi, Zhongzhi Yu, Yang Zhao, Yongan Zhang, Chaojian Li, Baopu Li, and Yingyan Lin. Vitcod: Vision transformer acceleration via dedicated algorithm and accelerator co-design. In The 29th IEEE International Symposium on High-Performance Computer Architecture (HPCA 2023), 2023. 1

[70] Guanghua Yu, Qinyao Chang, Wenyu Lv, Chang Xu, Cheng Cui, Wei Ji, Qingqing Dang, Kaipeng Deng, Guanzhong Wang, Yuning Du, et al. Pp-picodet: A better real-time object detector on mobile devices. arXiv preprint arXiv:2111.00902, 2021. 5, 6, 7

[71] Zhongzhi Yu, Yonggan Fu, Sicheng Li, Chaojian Li, and Yingyan Lin. Mia-former: Efficient and robust vision transformers via multi-grained input-adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 8962–8970, 2022. 2

[72] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 558–567, 2021. 2, 3

[73] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution vision transformer for dense predict. Advances in Neural Information Processing Systems, 34:7281–7293, 2021. 6

[74] 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. 13

[75] 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. 5

[76] Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, and Bryan Catanzaro. Long-short transformer: Efficient transformers for language and vision. Advances in Neural Information Processing Systems, 34, 2021. 1, 2

[77] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159, 2020. 3, 13

A. More Literature Review

Efficient ViTs. As previously mentioned in Sec. 2, efficient attention can be roughly categorized into two groups: local attention [2, 38, 56, 58] or kernel-based linear attention [1, 5, 6, 14, 29, 37, 39, 43, 66]. For local attention, Swin [38] restricts the window size of self-attention, so that only neighboring tokens will perform similarity measurements each other instead of all tokens; MaxViT [56] also adopts block attention within windows but additionally takes dilated global attention into account for learning both local and global information; QnA [2] shares the attention queries among all tokens; Linformer [58] approximates the queries and keys with low-rank factorization to reduce their vector length. For kernel-based linear attention, XCiT [1] proposes a "transposed" version of self-attention that operates across feature channels rather than tokens, resulting in linear complexity with the number of tokens; Linformer [58] explores a low-rank matrix to approximate the self-attention; Reformer [32] replaces self-attention by one that uses locality-sensitive hashing, changing its complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log(n))$ where n denotes the number of tokens; Longformer [3] combines a windowed local-context self-attention and a task-motivated global attention that encodes inductive bias about that task; Nystromformer [66] adapts the Nystrom method to approximate standard self-attention with $\mathcal{O}(n)$ complexity; Scatterbrain [8] unifies both low-rank approximation and sparse attention to improve accuracy-efficiency tradeoffs. Different from all above works, we explore from a new perspective by taking spectral angles into consideration when measuring the similarity among tokens, resulting in linear-angular attention with sparse training techniques that can achieve comparable or even better performance than softmax-based attention.

ViTs for Downstream Tasks. Apart from image classification tasks, ViTs have also been leveraged to serve as backbones for downstream tasks, such as object detection [7,34,60,74,77] and semantic segmentation [11,12,42]. For example, DETR [7] directly detects and predicts objects by combining a common CNN with a transformer architecture; Maskformer [12] proposes to use a simple mask classification model to predict a set of binary masks, each associated with a single global class label prediction. One big difference is that ViTs can beat CNNs on classification tasks that have a lower image resolution while are still less efficient than lightweight CNNs on downstream tasks that heavily rely on multi-scale resolution features. Therefore, there have been various debates on designing powerful ViT backbones: (1) *plain ViTs (e.g., ViTDet [33]) or hierarchical ViTs (e.g., MViTv2 [34], or Swin [38])?* Plain ViTs win in terms of simplicity but could be hard to scale down to lower resolution and computation regimes; Hierarchical ViTs seamlessly match with feature pyramid networks (FPNs) for extracting multi-scale feature maps but have more design factors to be considered or searched over. (2) *pure ViTs or hybrid CNN-ViTs?* Pure ViTs are compatible with self-supervised masked autoencoder (MAE) pretraining [25]; Hybrid CNN-ViTs can suffer from the information leakage problem [22] when adopting MAE pretraining, while being more efficient especially for feature extractions in early layers. Our proposed method does not fall into the aforementioned debates. Instead, it is compatible with all ViT variants relying on the softmax-based attention.

B. More Results and Clarification

Improvement from Our Training Recipe. Recall that in Sec. 4, we conduct experiments on three classical computer vision tasks. For object detection and semantic segmentation, we follow the baseline's training recipe for a fair and direct comparison. For the image classification, our training recipe has a minor difference due to the increased batch size and training epochs with more GPU nodes. As such, we further provide the detailed improvement breakdown here. Specifically, our adopted training recipe leads to ↑0.2% ∼ ↑1.6% top-1 accuracy improvements and our Caslting-ViT further reduces up to ↓40% MACs and increases ↑0.1% ∼ ↑1.2% top-1 accuracy simultaneously.

Ablation Studies on Image Classification. Our ablation studies are mostly done on the detection task as shown in Sec. 4.4 because of its less training time as compared to training ImageNet. Note that for these ablation studies, we do not adopt pretraining on ImageNet as specified in Sec. 4.4. After finishing the trial-and-error and when it comes to comparing with SOTA works, we then pretrain final models with the training recipe the same as LeViT [24], resulting in final results in Tab. 3. According to our experiments, training ImageNet takes nearly one week, while training COCO without pretraining on ImageNet takes only one day. In fact, ablation results on the classification task are consistent. To deliver more comprehensive ablation studies, we train Castling-LeViT-256 on ImageNet afterwards and find that: (1) + Lin.: 81.5%; (2) + Lin. & DWConv: 82.4%; (3) + Lin. & DWConv & SparseAttn: 82.6%, those results are consistent with our observation on detection experiments.

Conjecture of Why Linear-Angular Attention Sometimes Beats the Original Self-Attention. To better understand why the result of our Castling-ViT is even better than softmax-based ViTs. We summarize three differences between our method and previous linear attentions: (1) In addition to linear attention, we also take DWConv and sparse softmax-based attention into the training process; (2) We use a SGD optimizer instead of Adam, which is not common for training ViTs. Although Adam optimizer leads to faster convergence, we find that SGD optimizer helps to deliver better results if being trained sufficiently converged, e.g., we train 1000 epochs on ImageNet; (3) After revisit-

Table 8. Throughputs/memory measurements on a V100 for image classification, under various input resolutions denoted as $r \times r$.

Model	Throughputs (Images/s)			GPU Peak Memory (MB)		
	$r = 512$	$r = 1024$	$r = 1536$	$r = 512$	$r = 1024$	$r = 1536$
DeiT-Base	40	6	OOM	1220	12369	OOM
Castling-DeiT-Base	48 (↑20%)	8 (↑33%)	6	998 (↓18%)	4863 (↓61%)	15478
MViTv2-Base	43	5	OOM	1762	14686	OOM
Castling-MViTv2-Base	50 (↑16%)	10 (↑100%)	4	1483 (↓16%)	7028 (↓52%)	16028

Table 9. Ablation study on the patch size (measured on a V100).

Models	Throughputs (Images/s) under p patch sizes		
	$p = 8$	$p = 4$	$p = 2$
DeiT-Tiny	398	40	3
Castling-DeiT-Tiny	410 (↑1.0×)	103 (↑2.6×)	20 (↑6.7×)
DeiT-Base	60	8	OOM
Castling-DeiT-Base	64 (↑1.1×)	15 (↑1.9×)	4

Table 10. Latency measurements on a V100 for object detection.

Models	Params (M)	MACs (G)	mAP	Latency (ms)
YOLOv5-S	7.3	17.1	36.7	8.7 [23]
RetinaNet+PVT-Tiny [59]	23.0	221	36.7	-
Castling-ViT-L-416	13.1	5.3	37.3	3.9 (↓55.2%)

ing the attention design, we remove token/feature pooling and adopt post-**Q** pooling and residual connections [34] in our attention blocks. All above three differences contribute to the the final accuracy apart from the improvement of using linear-angular attention. We also show the breakdown analysis for each of these three points, see Sec. 4.4, Sec. B, and Sec. 3.1 for detailed analysis, respectively.

Actual Latency, Throughputs, and Memory Measurements. Our final models are dense and thus well compatible with GPUs. We measure and report the latency (↓55%), throughputs (↑16 ∼ ↑100%), and GPU memory (↓16 ∼ ↓61%) for both classification and detection tasks, as shown in Tab. 8/10. For throughputs, we measure both our Castling-ViT and baselines under their maximum allowed batch sizes (bs), i.e., bs=16/2/1, for different input resolutions r=512/1024/1536 in a fair and consistent V100 environment. Note that when the input resolution r=224, our models cannot beat the baseline in terms of throughputs because of (1) the newly added DWConv; (2) the removal of token/feature pooling. However, in terms of accuracy-efficiency tradeoffs, our Castling-ViT consistently beats all baselines as shown in Sec. 4. For memory, we record the peak memory per image. For latency, we benchmark with SOTA CNN-based detectors. Our model achieves 37.3mAP at 3.9ms latency on a V100, while YOLOv5-S only achieves 36.7mAP at 8.7ms latency). Moreover, Castling-ViT wins more throughputs (up to ↑6.7×) for smaller patch sizes and/or larger input resolutions, as shown in Tab. 9 and 8. Note that we record CUDA latency following the literature [38, 54]. All reported results are averaged among three runs.

Compare with ViT-based Baselines on Detection. We benchmark with SOTA CNN-based detectors under 6G MACs in Sec. 4 because that ViT-based detectors are too expensive. For example, our Castling-ViT achieves 37.3mAP at 5.3G MACs, while RetinaNet+PVT-Tiny only achieves 36.7mAP at even 221G MACs [59], as shown in 10.

Advantages of Angular Kernels? Angular kernels take into account extra spectral characteristics and enjoy good properties, e.g., positive semi-definite function → inner product in a high-dimensional and rich feature space, as analyzed in Sec. 3.2. It also achieves comparable accuracy with vanilla attention as validated by Sec. 4.

Large-Scale Ablation Studies on Attention Design. We use small ViTs for idea validation in Tab. 5 and the conclusion generalizes to larger ones. Here we add another ablation study on a larger model LeViT-384 as shown in Tab. 11, from which we see that the attention design insights consistently generalize from small models to larger models, further validating our design insights.

Why More Parameters Than Others in Low MACs? ViTs tend to have more parameters than CNNs under small MACs, e.g., LeViT [24] and Efficient-ViT [6]. For the LeViT, it features more layers with gradually downsampled input resolutions. For example, LeViT-256 requires 18.9M parameters at only 1.1G MACs, LeViT-384 requires 39.1M parameters at 2.4G MACs. Since we adopt LeViT-like structure to construct our Castling-ViT on image classification tasks, the parameter looks higher than other else baselines. Also, as indicated in Sec. 3.1, Castling-LeViT uses merely post-**Q** pooling, causing slightly higher hidden dimensions for **Q/K** than LeViT. In this work, we focus more on the FLOPs/latency instead of parameters since storage is not a major concern in modern hardware [48].

Will Auxiliary Attention and DWConv Work for Existing Linear Attentions? Yes, we train a DeiT-Tiny (w/o distill.; Acc.: 72.2%) w/ linear attention [17] for 300 epochs and observe that: (1) + Lin.: 68.3%; (2) + Lin. & DWConv: 71.7%; (3) + Lin. & SparseAttn: 70.2%; (4) + Lin. & DW-Conv & SparseAttn: 72.4%.

Clarify ReLU-S vs. ReLU-E in Tab. 5. During approximation, i.e., $\mathrm{Sim}(\mathbf{Q}, \mathbf{K}) \approx \phi(\mathbf{Q})\phi(\mathbf{K})^T$, both of them use ReLU as $\phi(\cdot)$, but ReLU-S takes the whole Softmax as $\mathrm{Sim}(\cdot)$, while ReLU-E takes the $\mathrm{Exp}(\cdot)$ as $\mathrm{Sim}(\cdot)$, e.g.,

Table 11. Additional ablation studies on attention designs using LeViT-384 on ImageNet.

Token	Feat.	Pre-Q	Post-Q	Residual Q	MACs (G)	Top-1 Accuracy (%)
✓		✓			2.50	82.63
	✓	✓			2.36	82.55
		✓			2.61	82.65
✓			✓	✓	2.83	82.19
	✓		✓	✓	2.80	81.86
			✓	✓	2.96	83.65

Efficient-ViT, resulting in additional divisions.