Mobile Attention: Mobile-Friendly Linear-Attention for Vision Transformers

Zhiyu Yao [1 2] **Jian Wang** [2] **Haixu Wu** [1] **Jingdong Wang** [2] **Mingsheng Long** [1]

Abstract

Vision Transformers (ViTs) excel in computer vision tasks due to their ability to capture global context among tokens. However, their quadratic complexity $\mathcal{O}(N^2D)$ in terms of token number N and feature dimension D limits practical use on mobile devices, necessitating more mobile-friendly ViTs with reduced latency. Multi-head linear-attention is emerging as a promising alternative with linear complexity $\mathcal{O}(NDd)$, where d is the per-head dimension. Still, more compute is needed as d gets large for model accuracy. Reducing d improves mobile friendliness at the expense of excessive small heads weak at learning valuable subspaces, ultimately impeding model capability. To overcome this efficiency-capability dilemma, we propose a novel Mobile-Attention design with a head-competition mechanism empowered by information flow, which prevents overemphasis on less important subspaces upon trivial heads while preserving essential subspaces to ensure Transformer's capability. It enables linear-time complexity on mobile devices by supporting a small per-head dimension d for mobile efficiency. By replacing the standard attention of ViTs with Mobile-Attention, our optimized ViTs achieved enhanced model capacity and competitive performance in a range of computer vision tasks. Specifically, we have achieved remarkable reductions in latency on the iPhone 12. Code is available at https://github.com/thuml/MobileAttention.

1. Introduction

Vision Transformers (ViTs) have achieved notable success in various computer vision applications, such as image classification and object detection, due to their powerful self-attention modules that effectively capture global context (Touvron et al., 2021). However, as shown in Figure 1 (a), the quadratic complexity $\mathcal{O}(N^2D)$ of the attention module (N is the number of tokens and D is the dimension of features), which increases with the number of input tokens, restricts the runtime efficiency of ViTs (Liu et al., 2021). To overcome this limitation, researchers have proposed linear attention designs that can reduce the quadratic complexity of standard ViT attention. These linear attention designs can be classified into two categories: (i) ViTs with local attention, where the size of the attention window is limited (Liu et al., 2021), and attention queries are shared (Arar et al., 2022; Wang et al., 2022); (ii) ViTs with kernel-based linear attention, which approximates the nonlinear softmax function by decomposing it into separate kernel embeddings, allowing for a change in matrix computation order to reduce computational complexity (Wu et al., 2022).

The popularity of kernel-based linear attention, as shown in Figure 1, is due to its linear computational complexity of $\mathcal{O}(NDd)$, where d represents the per-head dimension. This makes it an ideal choice for mobile devices. Unlike local attention, linear attention can capture global context across features over tokens. However, it can be computationally expensive, especially with a large head dimension (d), while mobile devices have limited memory resources. The large matrix multiplication between the key and value in the head dimension requires significant processing and storage, leading to out-of-memory errors or slow speeds.

As depicted in Figure 1, reducing head dimensions d will result in lower latency and improved efficiency in linear attention. Therefore, reducing d is an effective approach to make linear attention more mobile-friendly, supported both theoretically and in practical deployment. However, a small per-head dimension may cause some heads to struggle in learning valuable subspaces due to limited feature capacity, potentially resulting in trivial representations. These trivial subspaces can dominate over useful subspaces, impacting the expressiveness of linear attention and the overall performance of linear ViTs.

To tackle these challenges, we introduce the **Mobile-Attention** mechanism, the first kernel-based linear-attention specifically tailored for mobile devices. Going beyond the kernel-based linear attention, it leverages a small number

This work was done when Zhiyu Yao was intern at Baidu VIS. [1]School of Software, BNRist, Tsinghua University, Beijing, China [2]Baidu VIS, Beijing, China. Correspondence to: Mingsheng Long <mingsheng@tsinghua.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

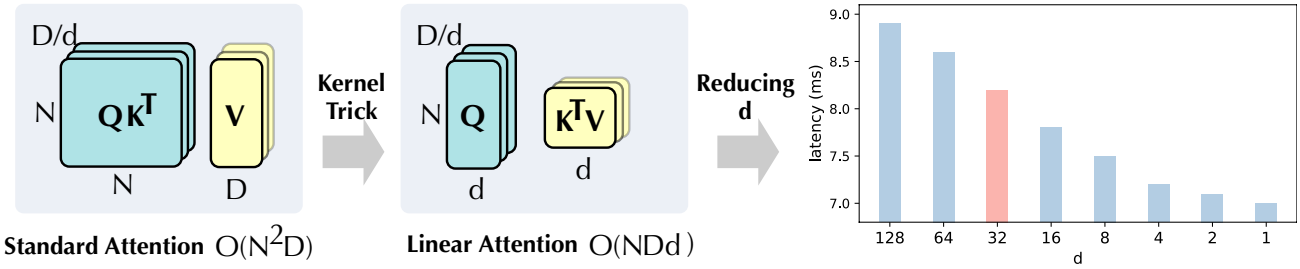


Figure 1. Standard attention in Transformers has a quadratic complexity with respect to the number of tokens N. Using a decomposable kernel allows rearranging operations so that linear attention scales with the square of features D, which is still not scalable in high dimensions. By reducing the head dimension d to approach 1, our Mobile-Attention achieves $\mathcal{O}(ND)$ complexity, resulting in lower latency and increased efficiency. The key challenge is to ensure model capability when using many small heads.

of head dimensions d and incorporates a head-competition mechanism to strengthen the model's capacity. This strategy effectively prevents overemphasis on less crucial subspaces upon trivial heads while retaining vital ones, ensuring an expressive and robust representation. By substituting the attention mechanism in ViTs by Mobile Attention, we present the **ViTs-MobileAtt** family, which is highly mobile-friendly and yields competitive performance across various vision tasks. In summary, we make the following contributions.

- We find that reducing the head dimensions d is vital for enhancing the mobile-friendliness of linear attention. However, this may lead to less effective heads, impacting their ability to learn valuable subspaces and potentially harming performance.
- We propose the *Mobile-Attention* mechanism, a new kernel-based linear attention for mobile devices to address the efficiency-capability dilemma. It supports lower head dimensions d for mobile-friendliness that were previously impossible, and a head-competition mechanism to guarantee the model capacity.
- Our optimized ViTs-MobileAtt, as new members to the ViTs family, offer improved mobility and competitive performance across various computer vision tasks. We have achieved remarkable reductions in latency. Specifically, under the prerequisite of retaining comparable or higher accuracy, we achieved a **25%** reduction in latency on DeiT, a **35%** reduction on PVTv2, and an **18%** reduction on EfficientformerV2.

2. Related Work

Vision Transformers. Vision Transformers (ViTs) (Dosovitskiy et al., 2020; Liu et al., 2021; Wu et al., 2021; Chu et al., 2021; Dong et al., 2021; Liu et al., 2022b; Zhai et al., 2022; Yao et al., 2023; Tu et al., 2022) have gained popularity in image classification due to its encoder-only transformer architecture that uses non-overlapping image patches as sequential inputs. However, this approach can be expensive as it requires pretraining on large datasets such as JFT300M (Sun et al., 2017). Fortunately, there are alternative approaches that can achieve comparable accuracy without costly pretraining. Strategies like DeiT (Touvron et al., 2021) and T2T-ViT (Yuan et al., 2021) employ improved training techniques and enhanced tokenization mechanisms, enabling them to attain ViT-like accuracy without pretraining on large datasets. In addition, recent works such as CrossViT (Chen et al., 2021), PVT (Wang et al., 2021), PVTv2 (Wang et al., 2022), and Swin-Transformer (Liu et al., 2021) propose pyramid-like architectures to improve the accuracy-efficiency tradeoffs of ViTs. While this architectural design is commonly used in CNNs (Howard et al., 2019), one of their main disadvantages is that they may require a large memory footprint, making it challenging to deploy on low-resource devices.

To enable better accuracy-efficiency tradeoffs, several efficient ViT architectures have been proposed to address the limitations of ViTs for resource-constrained devices, including LeViT (Graham et al., 2021), MobileViT (Mehta & Rastegari, 2021), and EfficientFormer (Li et al., 2022b). These architectures often reduce the number of attention layers, utilize channel-wise attention, or apply depth-wise convolutions to increase efficiency while maintaining accuracy. However, these models may sacrifice some accuracy for efficiency, and their performance may still fall behind state-of-the-art ViTs on large-scale datasets.

Kernel-based Linear-Attention Mechanisms. The concept of linear attention (Qin et al., 2022; Wu et al., 2022) is based on kernel functions that can approximate the softmax-attention mapping. Notably, Performers (Choromanski et al., 2020) use positive random features for softmax approximation, but this approach suffers from a significant performance drop due to limited modeling capacity. Other methods involve low-rank approximations such as SOFT (Lu

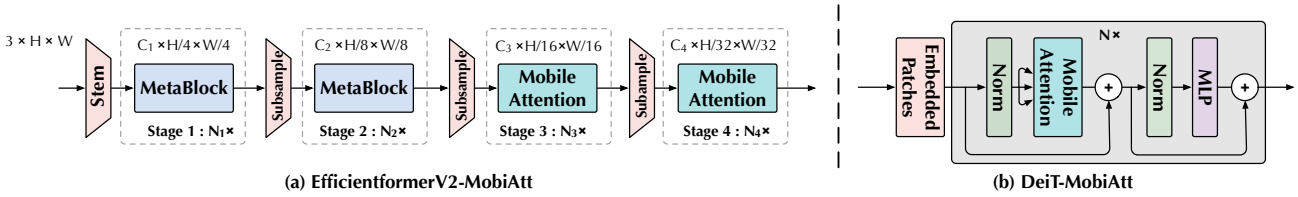


Figure 2. Network architectures for the ViTs-MobiAtt family. (a) We integrate our Mobile-Attention into the last two stages of EfficientformerV2 (Li et al., 2022a), creating EfficientformerV2-MobiAtt. The remaining modules are unchanged. (b) We substitute the standard attention in all transformer blocks of DeiT (Touvron et al., 2021) with our Mobile-Attention, resulting in DeiT-MobiAtt.

et al., 2021), the Nyström method (Xiong et al., 2021), and YOSO (Zeng et al., 2021), which approximate the softmax function using the Nyström technique. However, these approaches involve complex iterations in the calculations and are not compatible with the causal attention necessary for autoregressive tasks. Several studies (Hua et al., 2022; Zeng et al., 2022) have focused on improving both sparse and low-rank methods to attain fast attention matrix approximation. However, these methods may not be easily implemented on mobile devices like Apple CoreML. More recently, LARA (Zheng et al., 2022) recasts random-feature-based attention as self-normalized importance samplers. Nonetheless, its efficiency may not surpass other methods due to extra operations needed for importance sampling.

3. Preliminaries

Self-Attention Mechanism. The self-attention module serves as a crucial component in the Transformer (Vaswani et al., 2017) architecture, usually consisting of multiple heads. Each head collectively attends to information across various representation subspaces at distinct tokens, where N represents the number of tokens. Mathematically, self-attention can be expressed using the following equation:

$$\mathbf{A}_t^h = \sum_{i=1}^{N} \frac{\exp\left(\mathbf{Q}_t^h \mathbf{K}_i^{h\top}/\sqrt{d}\right)}{\sum_{j=1}^{N} \exp\left(\mathbf{Q}_t^h \mathbf{K}_j^{h\top}/\sqrt{d}\right)} \mathbf{V}_i^h, \tag{1}$$

where $t \in \{1, \cdots, N\}$, $h \in \{1, \cdots, M\}$ with N and M being the number of tokens and heads respectively. $\mathbf{A}_t^h$ refers to the t-th row of output from the h-th head's attention. The query, key, value vectors $\mathbf{Q}_t^h, \mathbf{K}_t^h, \mathbf{V}_t^h \in \mathbb{R}^d$ are obtained by multiplying the input $\mathbf{X}_t \in \mathbb{R}^D$ with three individual learnable weight matrices $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{D\times D}$, where D is the feature dimension and d is the per-head dimension. To calculate attention scores, each head in the self-attention module calculates the inner product between query-key pairs, then scales the products to stabilize training and normalize them using Softmax. This leads to a weighted sum of all value vectors. Once this is done for all attention heads, the resulting outputs are concatenated and a final linear projection using learnable weights is applied. It is important to note that the computation of Equation 1 has a quadratic complexity of $\mathcal{O}\left(N^2 D\right)$, making it difficult to use them on mobile devices without optimization techniques.

Kernel-based Linear Attention. The linear attention mechanism (Choromanski et al., 2020; Qin et al., 2022; Wu et al., 2022) is based on the concept of decomposing the similarity measure function into distinct kernel embeddings, such as $\exp\left(\mathbf{Q}_t \mathbf{K}_i^\top/\sqrt{d}\right) \approx \Phi(\mathbf{Q}_t)\Phi(\mathbf{K}_i)^\top$, so that Equation 1 will be reformulated as follows:

$$\begin{aligned} \mathbf{A}_t^h &= \sum_{i=1}^{N} \frac{\Phi\left(\mathbf{Q}_t^h\right) \Phi\left(\mathbf{K}_i^h\right)^\top}{\sum_{j=1}^{N} \Phi\left(\mathbf{Q}_t^h\right) \Phi\left(\mathbf{K}_j^h\right)^\top}, \\ \mathbf{V}_i^h &= \frac{\Phi\left(\mathbf{Q}_t^h\right) \sum_{i=1}^{N} \Phi\left(\mathbf{K}_i^h\right)^\top \mathbf{V}_i^h}{\Phi\left(\mathbf{Q}_t^h\right) \sum_{j=1}^{N} \Phi\left(\mathbf{K}_j^h\right)^\top}, \end{aligned} \tag{2}$$

where the multiplication of $\mathbf{Q}\mathbf{K}^\top$ is avoided by calculating the multiplication of keys and values firstly based on the associative law, specifically $\Phi(\mathbf{K})^\top\mathbf{V}$. Based on this, we can infer that the computational complexity is linear with respect to the sequence length, demonstrating a complexity of $\mathcal{O}\left(ND^2/M\right) = O\left(NDd\right)$. Nevertheless, this may still require substantial computational expenses, especially in the final Transformer stages where the number of head dimensions d is moderately large.

4. Methods

The proposed ViT-MobiAtt framework is based on Mobile-Attention, a novel linear attention method tailored for mobile devices. By supporting a substantially reduced number of head dimensions d and integrating a *head-competition* mechanism, Mobile-Attention significantly improves ViT's mobile-friendliness and enhances its model capacity.

Architecture for ViTs-MobiAtt. The Mobile-Attention is highly adaptable, allowing for seamless integration with various ViTs. By simply substituting the attention module with our Mobile-Attention, we can generate a variety of ViTs-MobiAtt without altering the underlying architecture. For instance, as illustrated in Figure 2, we apply the ViT-MobiAtt framework to two representative state-of-the-art

ViT architectures: the conventional vision transformer DeiT (Touvron et al., 2021), and the highly efficient transformer EfficientformerV2. This results in the architecture of DeiT-MobiAtt and EfficientformerV2-MobiAtt, respectively.

4.1. Motivation of Mobile-Attention

Our Mobile-Attention effectively captures the global context among tokens by incorporating a head-competition mechanism to further enhance the model capacity of each head, which is able to be more mobile-friendly and maintain the model capacity simultaneously.

Why to Reduce Head Dimensions? Our Mobile-Attention mechanism is based on the kernel-based linear attention approach described in Section 3. We utilize the non-linear function $\Phi(x) = \sigma\left(x/\|x\|_2\right)$, which combines the cosine similarity kernel with the sigmoid function σ. As discussed in Section 3, we discover that reducing the head dimensions d significantly lowers the computational complexity. Moreover, when d is decreased to a small constant (e.g., $d = 2$), the Mobile-Attention performs computationally-lightweight matrix multiplications. This reduction in matrix size is highly advantageous for mobile devices with limited memory resources, as it frees up space for other tasks and enhances overall device performance. In summary, lowering d effectively empowers linear attention to be more mobile-friendly, as demonstrated by both theoretical complexity and practical deployment. Thus, our Mobile-Attention utilizes extremely small head dimensions $d = 4$, resulting in nearly an $\mathcal{O}(ND)$ computational complexity, which is linear with respect to both token number length N and feature dimension D.

Head Competition is Necessary. Nevertheless, each head may encounter difficulties in learning meaningful subspaces due to limited feature capacity, potentially resulting in the acquisition of trivial information. These degenerated subspaces might dominate and overshadow the valuable ones, consequently affecting the overall performance of linear attention. To tackle this challenge, our Mobile-Attention attempts to incorporate a competition mechanism among heads. With competition, the trivial heads will be inhibited and the valuable information will be significantly highlighted, thereby ensuring that the valuable learned subspaces are not overwhelmed. This head-competition will promote effective learning and representation of crucial information, ultimately enhancing the performance and capacity of linear-attention based Transformer models. However, each head may function independently in the canonical multi-head attention (Vaswani et al., 2017), posing a challenge in constructing an effective head-competition mechanism.

Attention among Heads is Inefficient. A straightforward way to address the uninformative heads issue is to compute self-attention between $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ with respect to heads:

$$\mathbf{R}_t^h = \sum_{i=1}^{M} \frac{\exp\left(\mathbf{Q}_t^h {\mathbf{K}_t^i}^\mathsf{T}/\sqrt{d}\right)}{\sum_{j=1}^{M} \exp\left(\mathbf{Q}_t^h {\mathbf{K}_t^j}^\mathsf{T}/\sqrt{d}\right)} \mathbf{V}_t^i, \tag{3}$$

where $t \in \{1, \cdots, N\}$, $h \in \{1, \cdots, M\}$. This approach can communicate multiple heads and highlight valuable information with the softmax function. Still, similar to the self-attention among tokens, the head attention also suffers from high computational complexity $\mathcal{O}(NMD)$, especially when the head number is large, which contradicts the mobile-friendly goal.

Linear attention among heads can reduce the computation complexity with respect to the head number by decomposing the similarity measure function into distinct kernel embeddings, such as $\exp\left(\mathbf{Q}_t^h {\mathbf{K}_t^i}^\mathsf{T}/\sqrt{d}\right) \approx \Phi(\mathbf{Q}_t^h)\Phi(\mathbf{K}_t^i)^\mathsf{T}$. Consequently, Equation 1 can be reformulated as follows:

$$\begin{aligned} \mathbf{A}_t^h &= \sum_{i=1}^{M} \frac{\Phi\left(\mathbf{Q}_t^h\right) \Phi\left(\mathbf{K}_t^i\right)^\mathsf{T}}{\sum_{j=1}^{M} \Phi\left(\mathbf{Q}_t^h\right) \Phi\left(\mathbf{K}_t^j\right)^\mathsf{T}}, \\ \mathbf{R}_t^h &= \frac{\Phi\left(\mathbf{Q}_t^h\right) \sum_{i=1}^{M} \Phi\left(\mathbf{K}_t^i\right)^\mathsf{T} \mathbf{V}_t^i}{\Phi\left(\mathbf{Q}_t^h\right) \sum_{j=1}^{M} \Phi\left(\mathbf{K}_t^j\right)^\mathsf{T}}. \end{aligned} \tag{4}$$

We modify linear attention to communicate M different heads in linear time $\mathcal{O}(NDd)$. It is crucial to note that the kernel-based linear-attention is free from softmax function, leading to trivial attention weights that lose the advantage in highlighting important information.

4.2. Mobile-Attention Mechanism

Incoming Flow and Outgoing Flow. To accomplish the competition among heads efficiently, we adopt the concept of information flow from Flowformer (Wu et al., 2022), which reformulates the calculation of attention based on the incoming and outgoing flows between value vectors and final results. Orthogonal to Flowformer that tackles the long-sequence burden, in ViTs-MobiAtt, we focus on the head dimension. As depicted in Figure 3 (b), the incoming flow represents the global interaction between one result head and all value heads, whereas the outgoing flow captures the global interaction from one value head to all result heads. Under these concepts, we can calculate:

$$\begin{aligned} \mathbf{I}^h &= \Phi\left(\mathbf{Q}^h\right) \sum_{j=1}^{M} \Phi\left(\mathbf{K}^j\right)^\mathsf{T}, \\ \mathbf{O}^h &= \Phi\left(\mathbf{K}^h\right) \sum_{i=1}^{M} \Phi\left(\mathbf{Q}^i\right)^\mathsf{T}, \end{aligned} \tag{5}$$

which represent the capacity of incoming flow and outgoing flow, respectively, and $\mathbf{I}^h \in \mathbb{R}^{N\times 1}$, $\mathbf{O}^h \in \mathbb{R}^{N\times 1}$. The incorporation of the above flow capacities enable communication

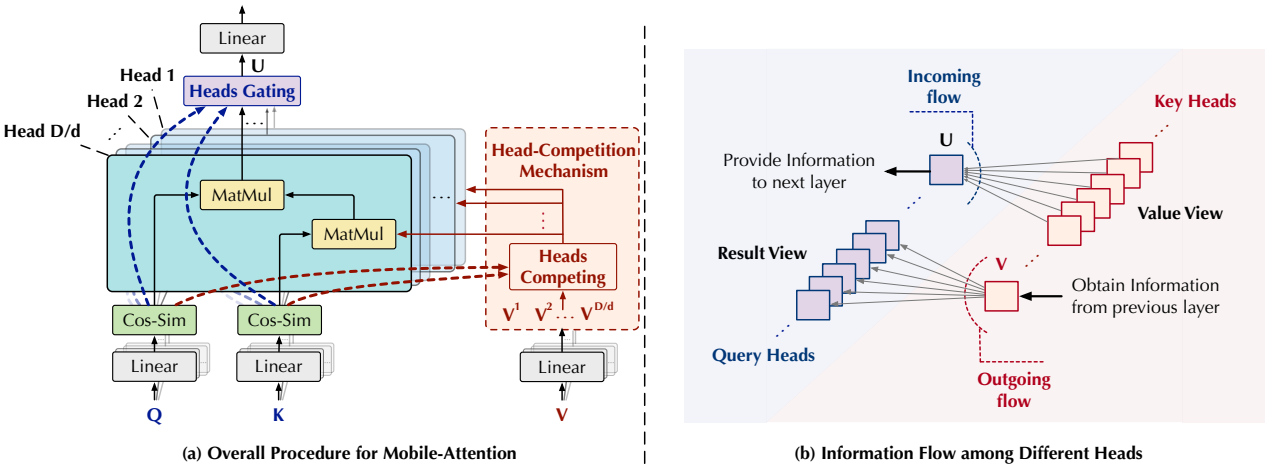


Figure 3. Procedure of Mobile-Attention (left); An information flow view among heads from values $\mathbf{V}$ to final results $\mathbf{U}$ (right).

across different heads, fostering competition and enhancing the model's expressiveness.

Head-Competition Mechanism. Our Mobile-Attention effectively develops a competition mechanism among heads by contrasting the capacity of incoming flow for final result tokens as 1, making the outgoing flow of value tokens compete with each other under this fixed sum situation. Also by contrasting the capacity of outgoing flow of values as 1, it can obtain the competed incoming flow of final results. In summary, based on Equation 2, we can compute the competed incoming flow and outgoing flow for each head based on the above calculated flow capacities as follows:

$$\begin{aligned}\overline{\mathbf{I}}^h &= \Phi\big(\mathbf{Q}^h\big)\sum_{h'=1}^{M}\frac{\Phi\big(\mathbf{K}^{h'}\big)^{\mathsf{T}}}{\mathbf{O}^{h'}},\\ \overline{\mathbf{O}} &= \Phi\big(\mathbf{K}^h\big)\sum_{h'=1}^{M}\frac{\Phi\big(\mathbf{Q}^{h'}\big)^{\mathsf{T}}}{\mathbf{I}^{h'}},\end{aligned} \tag{6}$$

where $\overline{\mathbf{I}}^h \in \mathbb{R}^{N\times 1}$, $\overline{\mathbf{O}}^h \in \mathbb{R}^{N\times 1}$, and $h \in \{1,\cdots,M\}$ denotes the capacity of competed incoming flow and outgoing flow of the h-th head respectively. These competed information flows are able to provide reliable evidence to present the importance of multiple heads.

Procedure of Mobile-Attention. As shown in Figure 3, we present the Mobile-Attention mechanism based on the above head-competed information flows, where competed outgoing flow of $\overline{\mathbf{O}}$ and competed incoming flow $\overline{\mathbf{I}}$ respectively indicate the importance of value heads and final result heads. By introducing $\overline{\mathbf{O}}$ and $\overline{\mathbf{I}}$ into the linear attention among tokens in Equation 2, the Mobile-Attention mechanism can naturally incorporate the competitive information among heads, which can be formalized as follows:

$$\begin{aligned}\overline{\mathbf{V}} &= \mathrm{Softmax}\left(\overline{\mathbf{O}}\right)\odot\mathbf{V},\\ \mathbf{U}_t^h &= \sigma\big(\overline{\mathbf{I}}_t^h\big)\frac{\Phi\left(\mathbf{Q}_t^h\right)\sum_{i=1}^{N}\Phi\left(\mathbf{K}_i^h\right)^{\mathsf{T}}\left(\overline{\mathbf{V}}_i^h\right)}{\Phi\left(\mathbf{Q}_t^h\right)\sum_{j=1}^{N}\Phi\left(\mathbf{K}_j^h\right)^{\mathsf{T}}},\end{aligned} \tag{7}$$

where $\odot$ denotes element-wise multiplication, σ represents the sigmoid function. $\overline{\mathbf{V}} \in \mathbb{R}^{M\times N\times d}$ is the value vector weighted by competed outgoing flow, where weights are assigned to each head, corresponding to *Heads Competing* in Figure 3. To compute the final result for each head and each token $\mathbf{U}_t^h$, we utilize the competed incoming flow $\overline{\mathbf{I}}$ to regulate each head's information, namely *Heads Gating* in Figure 3. Eventually, the final result $\mathbf{U} \in \mathbb{R}^{M\times N\times d}$ is obtained. Note that the calculations of competed information flow $\overline{\mathbf{I}}, \overline{\mathbf{O}}$ among heads and the linear attention among tokens are in the complexity of $\mathcal{O}(2ND)$ and $\mathcal{O}(NDd)$ respectively.

Following the canonical attention in Transformers (Vaswani et al., 2017), the find output of Mobile-Attention is obtained by applying a channel-dimension linear projection to $\mathbf{U}$. The above-mentioned designs, including competitive information flow among heads and linear attention among tokens, allow Mobile-Attention to foster competition among heads, preventing undesirable focus on less important details upon individual heads and preserving essential information across all heads. This competition design substantially enhances the model capacity and efficiency.

5. Experiments

To implement the ViT-MobiAtt framework, we utilized PyTorch 1.11, following common practices in recent research such as Swin Transformer (Liu et al., 2021) and T2t-ViT

Table 1. Classification results on ImageNet1k dataset compared with state-of-the-art model architectures, where EfficientformerV2-MobiAtt means that the idea of Mobile-Attention is applied on EfficientformerV2. The latency results are obtained by running models on iPhone 12 compiled with CoreMLTools, Pixel 6 (CPU) compiled with XNNPACK, and Nvidia A100 (GPU) compiled with TensorRT.

Model	Params(M)	GMACs	CoreML(ms)	A100 (ms)	Pixel 6 (ms)	Top-1 Acc(%)
MobileNetV2 (Sandler et al., 2018)	3.5	0.30	0.9	5.0	25.3	71.8
MobileViT-XS (Mehta & Rastegari, 2021)	2.3	0.70	7.3	11.7	64.4	74.8
EdgeViT-XXS (Chen et al., 2022)	4.1	0.60	2.4	11.3	30.9	74.4
EfficientNet-B0 (Tan & Le, 2019)	5.3	0.40	1.4	10.0	29.4	77.1
ConvNeXt-T (Liu et al., 2022a)	29.0	4.50	83.7	28.8	340.5	82.1
Swin-T (Liu et al., 2021)	29.0	4.50	97.3	22.0	-	81.3
DeiT-T (Touvron et al., 2021)	5.7	1.25	4.5	7.1	66.6	72.2
DeiT-T-MobiAtt	5.7	**1.22**	**3.8**	5.9	53.9	**73.3**
DeiT-S (Touvron et al., 2021)	22.0	4.60	9.0	15.5	218.2	79.8
DeiT-S-MobiAtt	22.0	**4.20**	**7.2**	13.3	175.7	**80.0**
DeiT-B (Touvron et al., 2021)	86.3	17.56	18.2	-	-	83.4
DeiT-B-MobiAtt	86.3	**17.03**	**13.3**	-	-	**84.2**
PVT-v2-b0 (Wang et al., 2022)	3.7	0.60	78.4	17.6	-	70.5
PVT-v2-b0-MobiAtt	**3.5**	**0.56**	**57.3**	15.0	-	**71.5**
PVT-v2-b2 (Wang et al., 2022)	25.4	4.00	101.0	36.2	-	82.1
PVT-v2-b2-MobiAtt	**21.1**	**3.80**	**65.6**	33.7	-	**82.6**
PVT-v2-b3 (Wang et al., 2022)	45.2	-	114.5	230.9	-	83.3
PVT-v2-b3-MobiAtt	**39.0**	-	**89.1**	210.1	-	**84.0**
EfficientFormerV2-S0 (Li et al., 2022a)	3.5	0.40	0.9	6.6	20.8	75.7
EfficientformerV2-S0-MobiAtt	3.5	0.37	**0.7**	5.5	16.2	**76.0**
EfficientFormerV2-S2 (Li et al., 2022a)	12.6	1.25	1.6	14.5	57.2	81.6
EfficientformerV2-S2-MobiAtt	12.6	**1.22**	**1.2**	13.1	48.9	**82.1**
EfficientFormerV2-L (Li et al., 2022a)	26.1	2.56	2.7	22.5	117.7	83.3
EfficientformerV2-L-MobiAtt	26.1	**2.50**	**2.2**	20.3	97.4	**83.7**

(Yuan et al., 2021). Our models were trained on a cluster of NVIDIA A100 GPUs to ensure optimal performance. Additionally, we measured the inference speed on mobile devices, specifically an iPhone 12 with an A14 Bionic chip running iOS version 15. These measurements were obtained by averaging over 1,000 runs and testing with both the neural processing unit (NPU) and the CPU. For deploying the runtime model with a batch size of 1, we used CoreMLTools. We also tested model latency on a Pixel 6 (Android) CPU. The models (batch size of 1) were compiled with XNNPACK to obtain latency metrics for most methods under comparison.

5.1. ImageNet-1K Classification

Settings. The image classification using the ImageNet dataset (Deng et al., 2009) with 1.2 million training and 50K validation images. In order to showcase the generalization capability of Mobile-Attention, we apply our Mobile-Attention mechanism to three popular vision transformers: DeiT (Touvron et al., 2021), PVT-v2 (Wang et al., 2022), and EfficientformerV2 (Li et al., 2022a), which is a state-of-the-art lightweight transformer model. Specifically, we replace only the attention blocks with our Mobile-Attention for these ViTs, while leaving other modules unchanged. For the classification task, we employ the AdamW optimizer (Loshchilov & Hutter, 2017) and train the model for 300 epochs. We set the batch size to 2048 and the learning rate to 0.001, using a cosine learning rate decay schedule. The resolution of the input image is resized to 224×224. All comparison models utilize the default number of heads according to their respective literature in previous work.

Generalizability of Mobile-Attention. As shown in Table 1, DeiT-MobiAtt and PVT-v2-MobiAtt consistently achieve higher performance and reduced latency on CoreML. For example, PVT-v2-b2-MobiAtt achieves a 0.5 point boost and a significant **36%** reduction in latency compared to PVT-v2-b2. Similarly, DeiT-S-MobiAtt achieves a 0.2 point increase and a notable **20%** reduction in latency compared to DeiT-B, while utilizing fewer GMACs. By leveraging Mobile-Attention, EfficientformerV2-MobiAtt consistently outperforms the original EfficientformerV2 in mobile friendliness and performance. For instance, EfficientformerV2-S2-MobiAtt achieves an impressive **82.1%** top-1 accuracy, surpassing Efficientformer-S2, and reduces latency on CoreML by **25%**. It is worth noting that Mobile-Attention not only achieves lower latency on the CoreML framework but also on GPUs like the A100. This showcases the efficiency of the Mobile-Attention across different platforms and highlights its excellent performance.

Table 2. Classification results of various linear-attention methods within a DeiT-S (Touvron et al., 2021) framework on the ImageNet1k dataset. We replace the standard attention in all blocks of DeiT-S with Mobile-Attention to form the DeiT-S-MobiAtt. DeiT-S-MobiAtt* denotes a version of the model that does not employ the head-competition mechanism.

Model	Complexity	GMACs	CoreML(ms)	Top-1 Acc (%)
Hydra-DeiT-S (Bolya et al., 2022)	$\mathcal{O}(ND)$	4.10	7.0	73.5
Castling-DeiT-S (You et al., 2023)	$\mathcal{O}(ND^2)$	4.52	9.4	79.8
DeiT-S (Touvron et al., 2021)	$\mathcal{O}(N^2D)$	4.60	9.0	79.8
DeiT-S-MobiAtt w/ vanilla design	$\mathcal{O}(ND^2)$	-	8.1	79.0
DeiT-S-MobiAtt* w/ SE (Hu et al., 2018)	$\mathcal{O}(ND^2)$	-	7.3	78.3
DeiT-S-MobiAtt* w/ GLU (Shazeer, 2020)	$\mathcal{O}(ND^2)$	-	7.3	77.5
DeiT-S-MobiAtt w/o Head-competing	$\mathcal{O}(ND)$	4.18	7.2	76.4
DeiT-S-MobiAtt	$\mathcal{O}(ND)$	4.20	**7.2**	**80.0**

Performance Comparison. Compared to the state-of-the-art CNN-based method ConvNeXt-T (Liu et al., 2022a), EfficientformerV2-MobiAtt achieves a higher **83.7%** top-1 accuracy and is significantly faster on CoreML. When integrated with other cutting-edge efficient transformers, our ViT-MobiAtt exhibits competitive performance. For example, EfficientformerV2-S0-MobiAtt outperforms EdgeViT-XXS (Chen et al., 2022) by a substantial margin of 1.6 points and is **3×** faster on CoreML. These results clearly demonstrate that our Mobile-Attention is highly mobile-friendly compared to other state-of-the-art methods, while maintaining high performance. In conclusion, these findings highlight the effectiveness of Mobile-Attention in enhancing the performance and efficiency of various vision transformers, showcasing its strong generalization capabilities. It is also noteworthy that our Mobile-Attention mechanism not only achieves lower latency on the CoreML framework but also demonstrates comparable efficiency on other mobile devices and GPUs, such as the Pixel 6 and A100. This indicates that Mobile-Attention is effective and efficient across various mobile platforms and GPU architectures, highlighting its versatility and competitive performance. The reduced latency observed in EfficientformerV2-L-MobiAtt across different mobile devices further underscores its practical applicability and efficiency in real-world scenarios.

Applying to Models with More Parameters. We have integrated our Mobile-Attention mechanism into larger models, specifically DeiT-B and PVT-v2-b3, which feature a higher parameter count. As demonstrated in Table 1, the modified versions, DeiT-B-MobiAtt and PVT-v2-b3-MobiAtt, show improved performance and efficiency compared to their original counterparts. This underscores the successful incorporation of our Mobile-Attention technique into various model architectures, highlighting its effectiveness in enhancing models with increased parameter counts.

Comparison with State-of-the-Art Linear Attention. Table 2 compares novel kernel-based linear-attention methods applied to the DeiT-S model. It is worth mentioning that our Mobile-Attention outperforms the latest state-of-the-art, Castling-DeiT-S model (You et al., 2023), with an accuracy of **80.0** compared to their 79.8. Furthermore, our DeiT-S-MobiAtt achieves lower latency with a value of **7.2**, while the Castling-DeiT-S model has a latency of 9.4. Moreover, when comparing our Mobile-Attention to hydra-attention (Bolya et al., 2023), we not only achieve better mobile efficiency but also superior performance. This is due to the competition mechanism present in our Mobile-Attention, which is lacking in hydra-attention and results in inferior performance. In conclusion, our Mobile-Attention is more mobile-friendly and suited for mobile devices than alternative linear-attention models. In addition, we also include the "vanilla design" in Equation 3 for comparison. We present the results of the "vanilla design" for DeiT, which utilizes the same number of heads as DeiT-S-MobiAtt. From the results, it is evident that DeiT employing the vanilla design with a higher number of heads achieves inferior performance compared to the original DeiT. This suggests that without the head-competition mechanism, the use of the vanilla design in DeiT cannot maintain model capacity.

Comparison with Channel Attention Mechanisms. We performed an additional ablation study on channel attention mechanisms. To ensure a fair comparison, we integrated the SENet (Hu et al., 2018) and GLU (Shazeer, 2020) into DeiT-S-MobiAtt while excluding the head-competition mechanism, resulting in DeiT-S-MobiAtt*. It is crucial to note that DeiT-S-MobiAtt* does not employ the head-competition mechanism. As shown, it is evident that the utilization of the head-competition mechanism is crucial for achieving improved performance and efficiency with Mobile-Attention. Both the SE and GLU mechanisms alone cannot guarantee the model capacity of DeiT-S. While DeiT-S-MobiAtt* may achieve better efficiency compared to the original DeiT-S, it experiences a performance decrease without the head-competition mechanism. This highlights the importance of our head-competition in maintaining optimal performance.

Table 3. Object detection & instance segmentation on MS COCO 2017 with the Mask RCNN pipeline. Semantic segmentation on the ADE20K dataset by using models as the feature encoder in Semantic FPN.

Backbone	CoreML (ms)	Params (M)	Detection & Instance Segmentation						Semantic
			AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	mIoU
PoolFormer-S12 (Yu et al., 2021)	18.2	12.0	37.3	59.0	40.1	34.6	55.8	36.9	37.2
EfficientFormer-L1 (Li et al., 2022b)	6.1	12.3	37.9	60.3	41.0	35.4	57.3	37.3	38.9
EfficientFormerV2-S2 (Li et al., 2022a)	6.5	12.6	43.4	65.4	47.5	39.5	62.4	42.2	42.4
EfficientformerV2-S2-MobiAtt	**4.7**	**12.0**	**43.7**	**65.6**	**47.7**	**39.8**	**62.6**	**42.5**	**43.1**
ResNet101 (He et al., 2016)	13.2	48.0	40.4	61.1	44.2	36.4	57.7	38.8	38.8
PoolFormer-S24 (Yu et al., 2021)	23.0	21.0	40.1	62.2	43.4	37.0	59.1	39.6	40.3
Swin-T (Liu et al., 2021)	486.5	29.0	42.2	64.4	46.2	39.1	64.6	42.0	41.5
EfficientFormer-L3 (Li et al., 2022b)	10.6	31.3	41.4	63.9	44.7	38.1	61.0	40.4	43.5
EfficientFormerV2-L (Li et al., 2022a)	10.3	26.1	44.7	66.3	48.8	40.4	63.5	43.2	45.2
EfficientformerV2-L-MobiAtt	**7.7**	**25.3**	**44.9**	**67.0**	**48.8**	**40.8**	**63.8**	**44.0**	**46.0**

5.2. Downstream Tasks

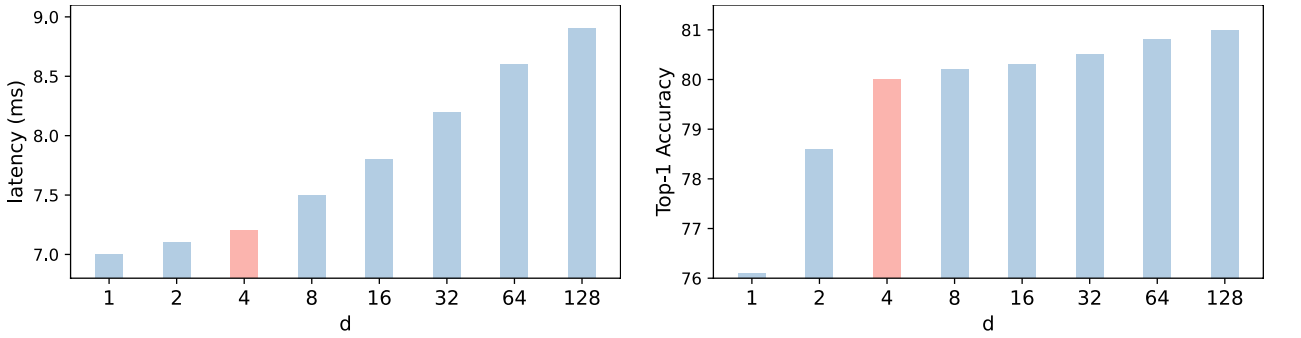
To demonstrate the superiority of our proposed Mobile-Attention mechanism, we evaluate its performance on two representative downstream vision tasks: Object Detection and Semantic Segmentation. Since some modules in Mask R-CNN (He et al., 2017) and Semantic FPN (Kirillov et al., 2019) are not easily deployable on CoreML, we assess only the backbone latency for a fair comparison, using the same test environments as in the classification task. For simplicity, we adopt a uniform input size of 512×512 for latency measurement, as shown in Table 3. To increase mobile device compatibility, we apply the Mobile-Attention mechanism to EfficientformerV2 (Li et al., 2022a) for downstream tasks, a state-of-the-art lightweight transformer. We replace only the attention blocks in the last two stages with our Mobile-Attention, keeping other modules unchanged, thereby creating the EfficientformerV2-MobiAtt.

Object Detection and Instance Segmentation. We adopt the Mask-RCNN implementation (He et al., 2017) to integrate EfficientformerV2-MobiAtt as the backbone and evaluate its performance. Our experiments are conducted on the COCO2017 dataset (Lin et al., 2014), which comprises 118K training images and 5K validation images. We initialize the EfficientformerV2-MobiAtt backbone with ImageNet-1K pretrained weights. Following previous work (Yu et al., 2021), we employ the AdamW optimizer (Loshchilov & Hutter, 2017) with an initial learning rate of 0.0002 and train the model for 12 epochs. The input size is set to 1333 × 800. The results for detection and instance segmentation are presented in Table 3. EfficientformerV2-MobiAtt consistently outperforms both CNN (ResNet) and transformer (PoolFormer) backbones. Under a similar computational budget, our EfficientformerV2-L-MobiAtt surpasses the ResNet101 backbone by 4.5 box AP and 4.4 mask AP, and outperforms the PoolFormer-S24 backbone by 4.8 box AP and 3.8 mask AP. These results demonstrate that EfficientformerV2-MobiAtt generalizes well as a robust backbone for various vision tasks.

Semantic Segmentation. To evaluate the performance of EfficientformerV2-MobiAtt in the semantic segmentation task, we conducted experiments using the challenging ADE20K dataset (Zhou et al., 2017). We built a semantic segmentation model by combining the EfficientformerV2-MobiAtt backbone with the Semantic FPN (Kirillov et al., 2019) segmentation decoder to ensure a fair comparison. The backbone was initialized with pretrained weights on ImageNet-1K, and the model was trained for 80K iterations using a total batch size of 32 across 8 GPUs. We adopted the AdamW optimizer (Loshchilov & Hutter, 2017) and implemented a polynomial learning rate schedule with a power of 0.9, starting from an initial learning rate of 0.0002. During training, we resized and cropped the input images to 512 × 512, and for testing on the validation set, we set the shorter side to 512, following common practices in segmentation.

As shown in Table 3, EfficientformerV2-MobiAtt consistently outperforms both CNN- and transformer-based backbones by a significant margin, while maintaining a similar computational budget. Notably, EfficientformerV2-MobiAtt achieves a mIoU improvement of **5.9** over PoolFormer-S12 (Yu et al., 2021). This result demonstrates that the Mobile-Attention mechanism in EfficientformerV2-MobiAtt enables better learning of long-term dependencies, which is advantageous in high-resolution dense prediction tasks.

Mobile Device Efficiency. As illustrated in Table 3, our EfficientformerV2-MobiAtt achieves the lowest latency time on CoreML among all baselines while maintaining competitive performance on object detection and semantic segmentation tasks. This result underscores that our Mobile-Attention is not only mobile-friendly but also exhibits strong performance in various downstream tasks, making it an ideal choice for deployment on mobile devices. Specifically, EfficientformerV2-L-MobiAtt is **3×** faster on CoreML



(a) Latency for different head dimensions of DeiT-S-MobiAtt **(b) Accuracy for different head dimensions of DeiT-S-MobiAtt**

Figure 4. A parameter analysis on ImageNet-1k is conducted for Mobile-Attention to evaluate the impact of per-head dimension d.

compared to PoolFormer (Yu et al., 2021) while outperforming it by a significant margin of **4.8** box AP, **2.2** mask AP, and **5.7** mIoU in detection, instance segmentation, and semantic segmentation tasks, respectively. In summary, our Mobile-Attention demonstrates a remarkable balance between efficiency and effectiveness, which sets it apart as a powerful solution for mobile vision applications.

5.3. Ablation Study

Effectiveness of Head-Competition Mechanism. The head-competition mechanism is essential to Mobile-Attention's performance. Table 2 demonstrates that without this mechanism, the model's effectiveness would be substantially diminished (**80.0%** $\rightarrow$ 76.4%). These findings underscore the importance of the competition mechanism among heads in balancing the number of heads and their influence on the model's performance.

Attention Visualiztion. Furthermore, we provide visualizations of the learned attention in Figure 5. Specifically, for DeiT-MobAtt, we visualize the competition weights $\mathrm{Softmax}(\overline{\mathbf{O}}) \in \mathbb{R}^{N\times 1}$, which are used for non-trivial aggregation of information from different sources. From the visualizations, we can observe that the Mobile-Attention mechanism is able to accurately capture the essential parts of the input. In contrast, without the head-competition mechanism, the hydra-attention (Bolya et al., 2023) fails to attend to the correct areas and produces a degenerated attention map. These visualizations further demonstrate the effectiveness of our proposed Mobile-Attention mechanism in improving attention capture and overall performance.

Reduced Per-Head Dimension is More Mobile-Friendly. As depicted in Figure 4 (a), the latency on CoreML decreases as the head dimensions d are reduced. This observation suggests that lowering the head dimensions effectively reduces latency, making our Mobile-Attention more mobile-friendly and suitable for deployment on devices with limited resources. On the other hand, as displayed in Figure 4 (b), the performance of DeiT-S-MobiAtt deteriorates when d diminishes, implying that an overly small d negatively impacts the model's capacity. To strike a balance between efficiency and performance, we ultimately select $d = 4$ as the optimal value. This choice ensures that our Mobile-Attention remains both effective and efficient for mobile devices.

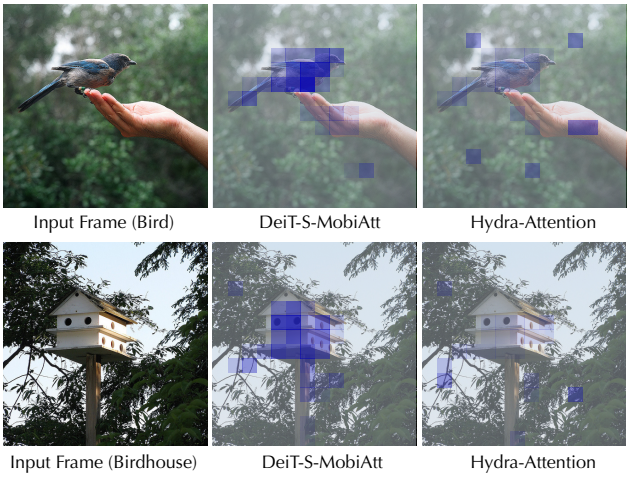


Figure 5. Visualization of learned attention. We present the sum of attention weights to each frame patch in the last layer.

6. Conclusion

Vision Transformers are prevalent for their self-attention modules, but quadratic complexity hinders efficiency. To address this efficiency-capability dilemma, we introduce Mobile-Attention, a mobile-friendly module that optimally balances attention heads while maintaining linear complexity. It employs a competition mechanism to enhance effectiveness of linear-attention, promoting competition among heads, thus preventing overemphasis on insignificant details and preserving essential information. Experiments show that ViTs-MobiAtt outperform state-of-the-art ViTs in classification, detection, and segmentation tasks on mobile devices, demonstrating superior efficiency and accuracy.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62022050 and U2342217), the BNRist Innovation Fund (BNR2024RC01010), and the National Engineering Research Center for Big Data Software.

Impact Statement

This paper presents a novel linear attention that empowers vision transformers for mobile devices. The potential societal consequences of this work depend on the particular application of computer vision tasks. Our method achieves excellent results in three real-world applications, which is beneficial for the entire research community. We are fully committed to ensuring ethical considerations are taken into account when developing our approach, and we believe there are no potential ethical risks associated with our work. We remain entirely focused on the scientific problem at hand.

References

Arar, M., Shamir, A., and Bermano, A. H. Learned queries for efficient local attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10841–10852, 2022.

Bolya, D., Fu, C.-Y., Dai, X., Zhang, P., and Hoffman, J. Hydra attention: Efficient attention with many heads. In *ECCV Workshops*, 2022.

Bolya, D., Fu, C.-Y., Dai, X., Zhang, P., and Hoffman, J. Hydra attention: Efficient attention with many heads. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pp. 35–49. Springer, 2023.

Chen, C.-F. R., Fan, Q., and Panda, R. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 357–366, 2021.

Chen, Z., Zhong, F., Luo, Q., Zhang, X., and Zheng, Y. Edgevit: Efficient visual modeling for edge computing. In *Wireless Algorithms, Systems, and Applications: 17th International Conference, WASA 2022, Dalian, China, November 24–26, 2022, Proceedings, Part III*, pp. 393–405. Springer, 2022.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., and Shen, C. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34:9355–9366, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*. IEEE, 2009.

Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., and Guo, B. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12114–12124, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., and Douze, M. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12259–12269, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:386–397, 2017.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.

Hua, W., Dai, Z., Liu, H., and Le, Q. Transformer quality in linear time. In *International Conference on Machine Learning*, pp. 9099–9117. PMLR, 2022.

Kirillov, A., Girshick, R. B., He, K., and Dollár, P. Panoptic feature pyramid networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6392–6401, 2019.

Li, Y., Hu, J., Wen, Y., Evangelidis, G., Salahi, K., Wang, Y., Tulyakov, S., and Ren, J. Rethinking vision transformers for mobilenet size and speed. *arXiv preprint arXiv:2212.08059*, 2022a.

Li, Y., Yuan, G., Wen, Y., Hu, J., Evangelidis, G., Tulyakov, S., Wang, Y., and Ren, J. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022b.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022a.

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., and Hu, H. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3202–3211, 2022b.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lu, J., Yao, J., Zhang, J., Zhu, X., Xu, H., Gao, W., Xu, C., Xiang, T., and Zhang, L. Soft: Softmax-free transformer with linear complexity. *Advances in Neural Information Processing Systems*, 34:21297–21309, 2021.

Mehta, S. and Rastegari, M. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021.

Qin, Z., Sun, W., Deng, H., Li, D., Wei, Y., Lv, B., Yan, J., Kong, L., and Zhong, Y. cosformer: Rethinking softmax in attention. *arXiv preprint arXiv:2202.08791*, 2022.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Shazeer, N. M. Glu variants improve transformer. *ArXiv*, 2020.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.

Tan, M. and Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. *ArXiv*, 2019.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., and Li, Y. Maxvit: Multi-axis vision transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pp. 459–479. Springer, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, pp. 5998–6008, 2017.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 568–578, 2021.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.

Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., and Zhang, L. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22–31, 2021.

Wu, H., Wu, J., Xu, J., Wang, J., and Long, M. Flowformer: Linearizing transformers with conservation flows. *arXiv preprint arXiv:2202.06258*, 2022.

Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y., and Singh, V. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Yao, T., Li, Y., Pan, Y., Wang, Y., Zhang, X.-P., and Mei, T. Dual vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

You, H., Xiong, Y., Dai, X., Wu, B., Zhang, P., Fan, H., Vajda, P., and Lin, Y. Castling-vit: Compressing self-attention via switching towards linear-angular attention during vision transformer inference. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2023)*, 2023.

Yu, S., Chen, T., Shen, J., Yuan, H., Tan, J., Yang, S., Liu, J., and Wang, Z. Unified visual transformer compression. *arXiv preprint arXiv:2203.08243*, 2022.

Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., and Yan, S. Metaformer is actually what you need for vision. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10809–10819, 2021.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.-H., Tay, F. E., Feng, J., and Yan, S. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 558–567, 2021.

Zeng, Z., Xiong, Y., Ravi, S., Acharya, S., Fung, G. M., and Singh, V. You only sample (almost) once: Linear cost self-attention via bernoulli sampling. In *International conference on machine learning*, pp. 12321–12332. PMLR, 2021.

Zeng, Z., Pal, S., Kline, J., Fung, G. M., and Singh, V. Multi resolution analysis (mra) for approximate self-attention. In *International Conference on Machine Learning*, pp. 25955–25972. PMLR, 2022.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12104–12113, 2022.

Zheng, L., Wang, C., and Kong, L. Linear complexity randomized self-attention mechanism. In *International Conference on Machine Learning*, 2022.

Zheng, L., Yuan, J., Wang, C., and Kong, L. Efficient attention via control variates. *ArXiv*, abs/2302.04542, 2023.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

A. Experiment on Head Pruning

We carried out a pruning experiment to demonstrate the effectiveness of our head-competition mechanism in ensuring that essential subspaces learned by specific heads are not overshadowed by trivial subspaces acquired by other heads. The competition performance achieved by the model after pruning a large number of trivial heads serves as evidence that the remaining valuable subspaces have a significant impact and are not overwhelmed by the trivial subspaces.

Setting. Specifically, we conducted a pruning experiment, following the approach established in previous literature on UVC (Yu et al., 2022). Initially, as described in Equation 5 of the main text, we computed the average score for each token associated to $\overline{\mathbf{O}}$: $\mathfrak{O} = \frac{1}{N}\sum_{i=1}^{N}\overline{\mathbf{O}}_i \in \mathbb{R}^{M\times 1}$, where N is the number of tokens, and M is the number of heads. $\mathfrak{O}^h \in \mathbb{R}$ represents the score after pruning the h-th head. Our technical approach concentrates on incorporating neuron-level and attention-head-level pruning in conjunction with knowledge distillation. We accomplish this by ranking the pruning scores $\{\mathfrak{O}^1, \mathfrak{O}^2, \cdots, \mathfrak{O}^M\}$ and eliminating the smallest 50% of attention heads based on their respective pruning scores. We integrate this pruning strategy into DeiT-MobiAtt, keeping the other pruning policies unchanged as in UVC (Yu et al., 2022).

Effectiveness of Head-Competition Mechanism. As demonstrated in Table 4, our method outperforms the advanced pruning approach UVC (Yu et al., 2022) (**79.0** vs. 78.5) in terms of performance. Additionally, our pruned DeiT-MobiAtt considerably exceeds the results obtained from random head pruning. These observations strongly suggest that our $\mathfrak{O}$ functions as an effective measure of each head's importance, ultimately contributing to more efficient pruning. We also observed that our Mobi-DeiT-S exhibits inferior performance without the head-competition mechanism, even when employing the cutting-edge pruning method UVC (Yu et al., 2022) (78.5 vs. 75.8). This observation highlights the critical role of the head-competition mechanism in preventing effective subspaces from being overshadowed by trivial ones. In the absence of such a competition incentive, the model fails to strike a balance between valuable and trivial subspaces, ultimately leading to inadequate learning of the remaining subspaces and resulting in worse performance.

Table 5. Semantic segmentation on the ADE20K dataset by using models as the feature encoder in Semantic FPN.

Model	CoreML (ms)	mIoU
EfficientFormerV2-S2 w/ EVA	6.8	41.0
EfficientFormerV2-S2 w/ Flow-Attention	6.6	38.4
EfficientFormerV2-S2	6.5	42.4
EfficientFormerV2-S2-MobiAtt	**4.7**	**43.1**

B. Linear-Attention for Downstream Tasks

Since linear-attention mechanisms have not been widely adopted for downstream tasks in computer vision, we conducted additional ablation studies to assess the effectiveness of existing state-of-the-art linear-attention mechanisms. Specifically, we performed experiments on semantic segmentation using the ADE20K dataset (Zhou et al., 2017). The experimental setup is consistent with the one described in Section 5.2 of the main text. We replaced the attention block of EfficientformerV2-S2 with the respective linear-attention mechanisms under comparison to establish a fair baseline. As shown in Table 5, our DeiT-S-MobiAtt, based on our Mobile-Attention, outperforms other linear-attention methods. For example, DeiT-S-MobiAtt achieves a 2.1-point improvement compared to the state-of-the-art linear-attention method EVA (Zheng et al., 2023). Regarding mobile efficiency, we also calculated the latency on Apple CoreML. Our ViT-MobiAtt not only achieves lower latency compared to other linear-attention methods but also demonstrates a more mobile-friendly design, making it a preferred choice for applications on mobile devices. The superior performance of our Mobile-Attention demonstrates its suitability for computer vision downstream tasks.

Table 4. Classification results for the pruning experiment within a DeiT-S (Touvron et al., 2021) framework on the ImageNet1k dataset. We replace the standard attention in all blocks of DeiT-S with Mobile-Attention to form the DeiT-S-MobiAtt. DeiT-S-MobiAtt w/ pruning score means using the $\mathfrak{O}$ to prune the heads. DeiT-S-MobiAtt* means the DeiT-S-MobiAtt trained without head-competition mechanism.

Model	GMACs	GMACs remained (%)	Top-1 Acc (%)
MobileNetV2 (Sandler et al., 2018)	0.30	100	71.8
DeiT-S-MobiAtt	4.20	100	**80.0**
DeiT-S-MobiAtt* w/ UVC (Yu et al., 2022)	2.13	50.6	75.8
DeiT-S-MobiAtt w/ UVC (Yu et al., 2022)	2.13	50.6	78.5
DeiT-S-MobiAtt w/ random pruning	2.13	50.6	76.9
DeiT-S-MobiAtt w/ pruning score	2.13	50.8	79.0