EleViT: exploiting element-wise products for designing efficient and lightweight vision transformers

Uzair Shah[1] , Jens Schneider[1], Giovanni Pintore[2], Enrico Gobbetti[2], Mahmood Alzubaidi[1], Mowafa Househ[1], Marco Agus[1]

[1] ICT Division, College of Science and Engineering, Hamad Bin Khalifa University Doha (Qatar)

(magus|jeschneider)@hbku.edu.qa

[2] Visual and Data-intensive Computing, CRS4, Italy

National Research Center in HPC, Big Data, and Quantum Computing, Italy

(giovanni.pintore|enrico.gobbetti)@crs4.it

Abstract

We introduce EleViT*, a novel vision transformer optimized for image processing tasks. Aligning with the trend towards sustainable computing,* EleViT *addresses the need for lightweight and fast models without compromising performance by redefining the multihead attention mechanism by primarily using element-wise products instead of traditional matrix multiplication. This modification preserves attention capabilities, while enabling multiple multihead attention blocks within a convolutional projection framework, resulting in a model with fewer parameters and improved efficiency in training and inference, especially for moderately complex datasets. Benchmarks against state-of-the-art vision transformers showcase competitive performance on low-data regime datasets like CIFAR-10, CIFAR-100, and Tiny-ImageNet-200.*

1. Introduction

The introduction of the transformer architecture [30] marked a paradigm shift in Natural Language Processing (NLP) and quickly transcended its original domain. By transforming image patches into tokens at various scales and incorporating positional encoding to capture spatial relationships, vision transformers (ViT) [9] have achieved impressive results on a variety of vision tasks. However, the computational and storage costs of their attention mechanism pose important challenges, especially in terms of training efficiency and deployment on resource-constrained environments such as single-GPU workstations or mobile architectures (Sec. 2).

Inspired by the natural accommodation mechanism in human vision, we introduce an efficient, full-fledged transformer architecture that replaces the conventional dot product in attention with an element-wise Hadamard product, akin to a blending process for focusing on foreground objects (Sec. 3.1). We integrate this mechanism into a novel architecture with multiple convolutional attention stages (Sec. 3.2), achieving efficient spatial attention across different channels. This mechanism, compatible with standard transformer architectures (Appendix A), offers training and inference efficiency benefits without compromising accuracy (Fig. 1). Our benchmarks demonstrate competitive performance in moderate-complexity dataset under constrained resource scenarios (Sec. 4). We present detailed comparisons with recent vision transformer models [18, 25] on CIFAR-10, CIFAR-100, and Tiny-ImageNet dataset, and include results from ablation studies.

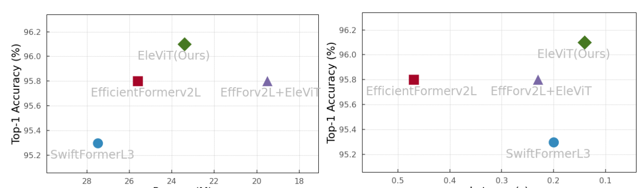

Figure 1. EleViT provides competitive accuracy as a function of parameter count (left) and latency time (right). We compare against SwiftFormer [25], EfficientFormer [18] and EfficientFormer modified with the proposed attention mechanism (Appendix A) on CIFAR10.

2. Related work

Vision transformers are the subject of extensive research. A full review is beyond the scope of this paper, and we refer the reader to recent surveys [6, 12, 15, 22] for a general coverage. Here, we focus on the solutions most closely related to our work, targeting the reduction of resources required for training and inference. Since the quadratic time and space complexity in the length of the sequence of the original attention mechanism in (Vision) Transformers [9, 30] is the major bottleneck, different strategies have been proposed to reduce it, with the dual goal of improving performance and reducing the need for training with extremely large numbers of examples.

Reformer [16] reduces the attention complexity by replacing the dot-product with one using locality-sensitive hashing and reversible residual layers instead of the standard residuals. Separable Vision Transformers [17] reduce the complexity in the local-global interaction within and among the windows in sequential order through a depth-wise separable self-attention. The Hierarchy Aware Feature Aggregation framework (HAFA) [5] improves the ConvNet feature aggregation scheme by adaptively enhancing the extraction of local features in shallow layers where semantic information is weak while aggregating patches with similar semantics in deep layers. Finally, recent architectures tried to reduce the quadratic complexity of the attention mechanism by reformulating it linearly. SwiftFormer [25] introduces an efficient additive attention mechanism replacing the quadratic matrix multiplication operations with linear element-wise multiplications. AFF [14] uses the Fourier Transform to convert the latent representation to the frequency domain and to perform filtering via an element-wise multiplication. Our architecture follows this trend by hybridizing the linearization process using element-wise products in the filtering stage for composing the attention values with the similarity weights obtained through the element-wise product of query and key components. As a result, we obtain an efficient method that provides good generalization without requiring extremely large data.

In addition to direct optimizations and reformulations of the attention mechanisms, many different orthogonal solutions have also been introduced to optimize vision transformers. The proposed approaches range from modeling multiple-scale attention in a way to separate the handling of local and global features [4, 7, 8, 13, 19, 29, 31], neural architecture search methods for optimizing hyperparameters and reducing training costs [2, 3, 11, 18, 21, 36], pruning strategies for tokens and coefficients [1, 10, 24, 28, 35], as well as the exploitation of quantization and mixed-precision components to reduce the size and improve caching behaviors [20, 23, 33]. These methods are orthogonal to ours, which can coexist with many of these other optimizations.

3. Method

Our efforts have been directed towards the creation of a more efficient and accurate vision transformer through the definition of a novel simplified attention mechanism (Sec. 3.1) and the design of architecture around it (detailed in Sec. 3.2 and depicted in Fig. 2). This refined model aims to enhance the generalization capacity when dealing with constrained training data while concurrently mitigating memory and computational complexity.

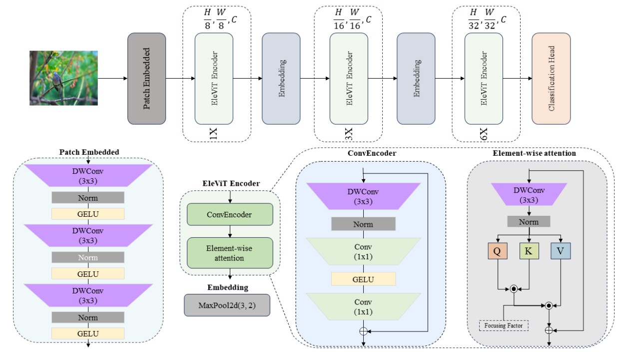

Figure 2. **EleViT architecture:** The proposed architecture is designed around the element-wise attention mechanism, and it features four stages, composing convolutional layers, residual connections, and batch normalizations in a bottleneck fashion.

3.1. Proposed Attention mechanism

Our methodology entails the application of a 3×3 convolutional projection to the input image, yielding query, key, and value representations. By concurrently utilizing 3D tensors, our approach captures global context within each spatial dimension per channel. The number of channels serves as distinct heads capturing diverse visual representations. This perspective enhances the model's comprehension of intrinsic relationships within the data, enabling the interplay of Q and K within a given channel and across spatial locations. To this end, the proposed mechanism is akin to a blending operator, and it better emulates the accommodation process employed in a vision for improving the visual clarity of foreground objects (see Fig. 3 in the Appendix). The channel-wise spatial-attention mechanism operates by taking an input $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$. This input comprises C channels within an image and possesses height H and width W. To process this input, $\mathbf{X}$ undergoes a transformation, generating $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ representations through the application of three distinct convolutional filters: W_q, W_k, and W_v where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{C\times H\times W}$. In general, there are various ways to process the representations to produce similarity scores [26]. In our case we considered the (Hadamard) element-wise product, already used successfully for neural question answering systems [32]: the similarity scores are computed between the $\mathbf{Q}$ and $\mathbf{K}$, and then passed through the softmax layer to obtain attention weights $\mathbf{F}$ ranging between 0 and 1. We also consider an additional learnable hyperparameter focusing factor α to further modulate the attention weights, according to the following equation:

$$\mathbf{F}_{[B,C,H,W]} = \alpha \cdot \text{softmax}\left(\frac{\mathbf{Q}_{[B,C,H,W]} \odot \mathbf{K}^T_{[B,C,H,W]}}{\sqrt{C}}\right) \tag{1}$$

Finally, the attention weights $\mathbf{F}$ are element-wise multiplied with the corresponding $\mathbf{V}$, defined as follows,

$$\overline{\mathbf{X}}_{[B,C,H,W]} = \mathbf{F}_{[B,C,H,W]} \odot \mathbf{V}_{[B,C,H,W]} \tag{2}$$

to obtain the self-attention representation $\overline{\mathbf{X}}$.

3.2. EleViT Architecture

EleViT draws inspiration from recent advances in hybrid architectures, particularly SwiftFormer [25] and EfficientFormerV2 [18]. It revolves around utilizing the channel-wise self-attention mechanism introduced above, wherein element-wise attention is conducted independently for each channel. We implement depth-wise transformations to derive the $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ tensors, with a key emphasis on enhancing computational efficiency. As illustrated in Fig. 2, EleViT employs a three-stage hierarchical design, obtaining feature sizes of $\{\frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$ of the input resolution. Similar to EfficientFormerV2 and SwiftFormer, EleViT commences with a 3×3 kernel convolution with a stride of 2 in the *stem* to embed the input image. We downsample the input tensor to $\frac{1}{8}$ instead of $\frac{1}{4}$ to reduce latency. Each convolution layer is followed by batch normalization and GELU activation. Stem is defined as:

$$\overline{\mathbf{X}}_{[B,C,\frac{H}{8},\frac{W}{8}]} = \text{stem}\left(\mathbf{X}_{[B,3,H,W]}\right) \tag{3}$$

where B denotes the batch size, C refers to the channel of the tensor, H and W are the height and width of the feature $\overline{\mathbf{X}}$. whereas $\overline{\mathbf{X}}$ is the output patch embed while $\mathbf{X}$ is the input image. In the subsequent three stages, we utilize an EleViT encoder consisting of ConvEncoder and element-wise attention. The architecture maintains consistency, featuring ConvEncoder followed by element-wise attention.

ConvEncoder Our ConvEncoder, akin to SwiftFormer with slight modifications in point-wise convolution, increases the number of input channels fourfold in the first layer and reduces it back to the input channels in the second layer. The input feature maps $\overline{\mathbf{X}}_i$ are fed into a 3x3 convolution (DWconv) followed by batch normalization (BN). The resulting features are fed to two pointwise convolutions (Conv1) alongside GELU activation. Finally, a residual connection is incorporated to facilitate information flow across the network. The ConvEncoder is defined as follows,

$$\overline{\mathbf{X}}_o = \text{Conv}_1(\text{Conv}_{1,G}(\text{DWconv}_{BN}(\overline{\mathbf{X}}_i))) + \overline{\mathbf{X}}_i \tag{4}$$

where $\overline{\mathbf{X}}_i$ is the input feature, $\overline{\mathbf{X}}_o$ is the output, $\text{Conv}_{1,G}$ is a 1×1 convolution with GELU activation, DWconv_{BN} is a 3×3 convolution with batch normalization.

Element-wise attention As illustrated in Fig. 2, our element-wise attention takes the input features $\overline{\mathbf{X}}_i$ and feeds them into three distinct 3×3 depth-wise convolutions (DWconv) followed by batch normalization (BN) to extract query, key, and value. The query is then element-wise multiplied with the transpose of the key to obtain the similarity scores [32], divided by the square root of the number of channels for smoothing, and passed through the softmax function to normalize the attention weights. Subsequently, the attention weights are multiplied by a scalar focusing factor (α), and the result is element-wise multiplied with the values for the final attention. Finally, a residual connection is introduced to enable information flow across the network. The element-wise attention is defined as follows:

$$\mathbf{Q} = \text{DWconv}_{BN}(\overline{\mathbf{X}}_i) \tag{5}$$

$$\mathbf{K} = \text{DWconv}_{BN}(\overline{\mathbf{X}}_i) \tag{6}$$

$$\mathbf{V} = \text{DWconv}_{BN}(\overline{\mathbf{X}}_i) \tag{7}$$

where $\overline{\mathbf{X}}_i$ is the input feature tensor, $DWconv_{BN}$ represents depth-wise convolution alongside with batch normalization. The query $\mathbf{Q}$ and key $\mathbf{K}$ undergo Eq. (1) to obtain focused attention weights $\mathbf{F}$, while the focused attention weights and $\mathbf{V}$ pass through Eq. (2).

4. Results

We evaluate our architecture with three low-data regime dataset: CIFAR100, CIFAR10, and Tiny-Imagenet-200. Implementation details are provided in Appendix A.

Tab. 1 compares our proposed EleViT model with state-of-the-art lightweight models, EfficientFormerV2 and SwiftFormer. The experiments involved training EfficientFormerv2l, SwiftFormerL3, EfficientFormerv2l+EleViT, and our proposed model using the same experimental setup across three distinct dataset: CIFAR100, CIFAR10, and TinyImageNet200. Inference latency measurements were conducted on a GeForce RTX 3090 with a batch size of 128.

CIFAR100 All models underwent training from scratch, utilizing an image resolution of 224×224. In the evaluation phase, EfficientFormerv2l+EleViT achieved a commendable top-1 accuracy of 81.2%, surpassing the benchmarks set by state-of-the-art lightweight models, EfficientFormerV2, and SwiftFormer. This achievement was particularly notable given that EfficientFormerv2l+EleViT demonstrated superior performance with 25% fewer parameters and a 2x faster inference speed than EfficientFormerV2. Furthermore, it approached the inference speed of SwiftFormer, a noteworthy accomplishment in the realm of lightweight model efficiency. These results underscore the effectiveness of our attention mechanisms in augmenting the overall efficiency of lightweight models. Addition-

dataset	Model	Lat. (s)↓	Param (M)↓	NAS	Top-1 (%)↑
CIFAR100	SwiftFormerL3	0.2	27.5	✗	72.6
	EfficientFormerv2L	0.47	25.6	✔	79.2
	EfficientFormerv2L+EleViT	0.23	**19.5**	✔	**81.1**
	EleViT (Our)	**0.14**	23.4	✗	79.7
CIFAR10	SwiftFormerL3	0.2	27.5	✗	95.3
	EfficientFormerv2L	0.47	25.6	✔	95.8
	EfficientFormerv2L+EleViT	0.23	**19.5**	✔	95.8
	EleViT (Ours)	**0.14**	23.4	✗	**96.1**
TinyImagenet200	SwiftFormerL3	0.2	27.5	✗	59.9
	EfficientFormerv2L	0.47	25.6	✔	**66.3**
	EfficientFormerv2L+EleViT	0.23	**19.5**	✔	64.2
	EleViT (Ours)	**0.14**	23.4	✗	64.8

Table 1. Comparison of model performance on CIFAR100, CIFAR10, and TinyImageNet200 dataset. Latency, parameter count (Param), Network Architecture Search (NAS), and Top-1 accuracy are provided for each model.

FF	Q-K	F-V	Acc @64	Acc @224	Latency
✔	$\cdot$	$\cdot$	90.49	94.87	0.16
	$\cdot$	$\odot$	91.03	94.94	0.15
	$\odot$	$\cdot$	90.82	94.96	0.15
	$\odot$	$\odot$	91.02	**95.22**	0.14
✗	$\cdot$	$\cdot$	90.78	94.67	0.16
	$\cdot$	$\odot$	**91.54**	94.65	0.15
	$\odot$	$\cdot$	90.78	94.38	0.15
	$\odot$	$\odot$	91.31	94.7	0.14

Table 2. Our ablation analysis compares different operator mechanisms´ impact on image classification accuracy for CIFAR10. The Focusing Factor (FF) denotes whether α is used; dot product $(\cdot)$ or element-wise multiplication $(\odot)$ denotes the operators used for attention computation in the query key (Q-K, see Eq. (1)) or focus-value (F-V, Eq. (2)) multiplication. The analysis begins with a resolution of 64x64 pixels (Acc @64) and further verification is conducted at the higher resolution of 224x224 pixels (Acc @224).

ally, EleViT, with its 23M parameters, exhibited competitive performance, outpacing EfficientFormerV2 and SwiftFormer by running $3\times$ and $1.5\times$ faster, respectively.

CIFAR10 Our proposed EleViT model achieved a top-1 accuracy of 96.1% over the test set, surpassing the performance of state-of-the-art lightweight models, namely EfficientFormerV2 and SwiftFormer. EleViT demonstrated efficiency with 20% and 10% fewer parameters and $3\times$ and $1.5\times$ faster inference speeds compared to EfficientFormerV2 and SwiftFormer, respectively. Moreover, Fig. 4 in the Appendix compares the validation losses tracked during training and shows that EleViT enables a more efficient training process, and the validation loss reaches its minimum in a lower number of epochs compared to the competitors. Furthermore, the hybrid model EfficientFormerV2+EleViT achieved performance parity with EfficientFormerV2, showcasing comparable accuracy with 25% fewer parameters and a $2\times$ increase in inference speed. This result underscores our attention mechanisms' efficacy in enhancing lightweight models' efficiency.

TinyImageNet200 EfficientFormerV2 emerged as a top performer in this rigorous setting, attaining a notable top-1 accuracy of 66.3% over the test set. EleViT, boasting fewer parameters and achieving a 3x faster inference speed, demonstrated compelling performance with a top-1 accuracy of 64.8%. Moreover, when integrated with our attention mechanism, EfficientFormerV2+EleViT achieved a commendable top-1 accuracy of 64.2%, comparable to the original EfficientFormerV2 while running $2\times$ faster. These results underscore EleViT's efficacy in achieving a balance between model efficiency and accuracy.

Ablation Study We employed consistent hyperparameters from Appendix A as defaults for all experiments. The ablation commenced with a 64×64 image resolution and progressed to 224×224 in subsequent trials for the CIFAR10 dataset. In the ablation reports, inference latency was measured on a GeForce RTX3090 GPU with a batch size 128. The number of epochs was selected based on Fig. 4, where the model loss reached its minimum at 60 epochs. The CIFAR10 ablation analysis highlights the role of the focusing factor α in enhancing image classification accuracy (Tab. 2). The choice between dot product $(\cdot)$ and element-wise multiplication $(\odot)$ in query-key (Q-K) and attention weights-value (F-V) computations significantly influences model outcomes. Element-wise multiplication consistently outperforms the dot product, capturing key-query relationships more effectively.

5. Conclusion

We have introduced an innovative vision transformer that streamlines the attention mechanism, using modulated element-wise products that emulate the natural vision process of foreground focus in scenes. Our model demonstrates promising results in low-regime dataset classification tasks. All experiments were conducted on workstations with standard, commercially available GPUs. Extending the benchmarking on larger dataset, such as ImageNet, in labs with more advanced computational capabilities forms an exciting future research direction. There is also a need for further experimentation to evaluate the effectiveness of our vision transformer in various image-to-image translation tasks, such as depth estimation [34] and semantic segmentation [27]. We plan to investigate this avenue in the near future.

References

[1] Maxim Bonnaerens and Joni Dambre. Learned thresholds token merging and pruning for vision transformers. *Transactions on Machine Learning Research*, 2023. 2

[2] Arnav Chavan, Zhiqiang Shen, Zhuang Liu, Zechun Liu, Kwang-Ting Cheng, and Eric P. Xing. Vision transformer slimming: Multi-dimension searching in continuous optimization space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4931–4941, June 2022. 2

[3] 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 (ICCV)*, pages 12270–12280, October 2021. 2

[4] Richard Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. In *International Conference on Learning Representations*, 2022. 2

[5] Yongjie Chen, Hongmin Liu, Haoran Yin, and Bin Fan. Building vision transformers with hierarchy aware feature aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5908–5918, October 2023. 2

[6] Krishna Teja Chitty-Venkata, Murali Emani, Venkatram Vishwanath, and Arun K. Somani. Neural architecture search for transformers: A survey. *IEEE Access*, 10:108374–108412, 2022. 1

[7] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34:9355–9366, 2021. 2

[8] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *European Conference on Computer Vision*, pages 74–92. Springer, 2022. 2

[9] 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

[10] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 396–414, Cham, 2022. Springer Nature Switzerland. 2

[11] Chengyue Gong, Dilin Wang, Meng Li, Xinlei Chen, Zhicheng Yan, Yuandong Tian, qiang liu, and Vikas Chandra. NASVit: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training. In *International Conference on Learning Representations*, 2022. 2

[12] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, Zhaohui Yang, Yiman Zhang, and Dacheng Tao. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):87–110, 2023. 1

[13] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11936–11945, October 2021. 2

[14] Zhipeng Huang, Zhizheng Zhang, Cuiling Lan, Zheng-Jun Zha, Yan Lu, and Baining Guo. Adaptive frequency filters as efficient global token mixers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6049–6059, October 2023. 2

[15] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Comput. Surv.*, 54(10s), sep 2022. 1

[16] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. 2

[17] Wei Li, Xing Wang, Xin Xia, Jie Wu, Xuefeng Xiao, Min Zheng, and Shiping Wen. Sepvit: Separable vision transformer. *arXiv preprint arXiv:2203.15380*, 2022. 2

[18] Yanyu Li, Ju Hu, Yang Wen, Georgios Evangelidis, Kamyar Salahi, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Rethinking vision transformers for mobilenet size and speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16889–16900, October 2023. 1, 2, 3, 7

[19] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 2

[20] Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repqvit: Scale reparameterization for post-training quantization of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17227–17236, 2023. 2

[21] Jihao Liu, Xin Huang, Guanglu Song, Hongsheng Li, and Yu Liu. Uninet: Unified architecture search with convolution, transformer, and mlp. In *European Conference on Computer Vision*, pages 33–49. Springer, 2022. 2

[22] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He. A survey of visual transformers. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2023. 1

[23] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34:28092–28103, 2021. 2

[24] 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

[25] Abdelrahman Shaker, Muhammad Maaz, Hanoona Rasheed, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Swiftformer: Efficient additive attention for transformer-based real-time mobile vision applications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17425–17436, October 2023. 1, 2, 3, 7

[26] Yuanyuan Shen, Edmund M-K Lai, and Mahsa Mohaghegh. Effects of similarity score functions in attention mechanisms on the performance of neural question answering systems. *Neural Processing Letters*, 54(3):2283–2302, 2022. 2

[27] Hengcan Shi, Munawar Hayat, and Jianfei Cai. Transformer scale gate for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3051–3060, June 2023. 4

[28] Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch slimming for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12165–12174, June 2022. 2

[29] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. *ECCV*, 2022. 2

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 2

[31] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention. In *International Conference on Learning Representations*, 2022. 2

[32] Dirk Weissenborn, Georg Wiese, and Laura Seiffe. Making neural qa as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, 2017. 2, 3

[33] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *European Conference on Computer Vision*, pages 191–207. Springer, 2022. 2

[34] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18537–18546, June 2023. 4

[35] Qingru Zhang, Simiao Zuo, Chen Liang, Alexander Bukharin, Pengcheng He, Weizhu Chen, and Tuo Zhao. Platon: Pruning large transformer models with upper confidence bound of weight importance. In *International Conference on Machine Learning*, pages 26809–26823. PMLR, 2022. 2

[36] Qinqin Zhou, Kekai Sheng, Xiawu Zheng, Ke Li, Xing Sun, Yonghong Tian, Jie Chen, and Rongrong Ji. Training-free transformer architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10894–10903, June 2022. 2

A. Implementation Details

To assess the performance of our model, we conducted comparative training sessions with EfficientFormerv2 and Swiftformer. The Python scripts for these models are directly extracted from the official GitHub repository. In the original architecture of EfficientFormerv2, the attention mechanism is strategically applied in the last two stages, whereas the original resolution in the second last stage is successively reduced to $\frac{H}{16}$ and $\frac{W}{16}$, and to $\frac{H}{32}$ and $\frac{W}{32}$. Subsequently, in the second last stage, the image is downsampled to the latter resolution for the attention layer, effectively reducing the model's complexity, and subsequently upsampled to $\frac{H}{16}$ and $\frac{W}{16}$. In our adaptation, EfficientFormerv2+EleViT, we introduce a modification by replacing the original attention mechanism with our proposed attention mechanism. Importantly, we maintain the original resolution in the second last stage of the architecture. This deliberate adjustment is made to assess the impact of our attention mechanism on the model's performance while preserving the resolution characteristics integral to the original EfficientFormerv2 architecture. This approach ensures a meticulous examination of the specific contribution of our attention mechanism, providing valuable insights into its effectiveness within the given context. All models undergo a comprehensive training regimen, initializing on each dataset for 150 epochs. This training employs an AdamW optimizer and incorporates a cosine learning rate scheduler, with the initial learning rate set to 1×10^{-3} unless explicitly specified otherwise. Throughout the training and testing phases, images are consistently maintained at a resolution of 224×224 pixels. These experimental procedures are meticulously implemented using PyTorch 2.1, with computations executed on a single NVIDIA GeForce RTX 3070 GPU. A batch size of 32 is carefully chosen for the training process. Moreover, data augmentation techniques, including random crop, random horizontal flip, 10-pixel cut-out, and cut-mix augmentation, are systematically applied, emphasizing image mixing, wherein two images are seamlessly integrated.

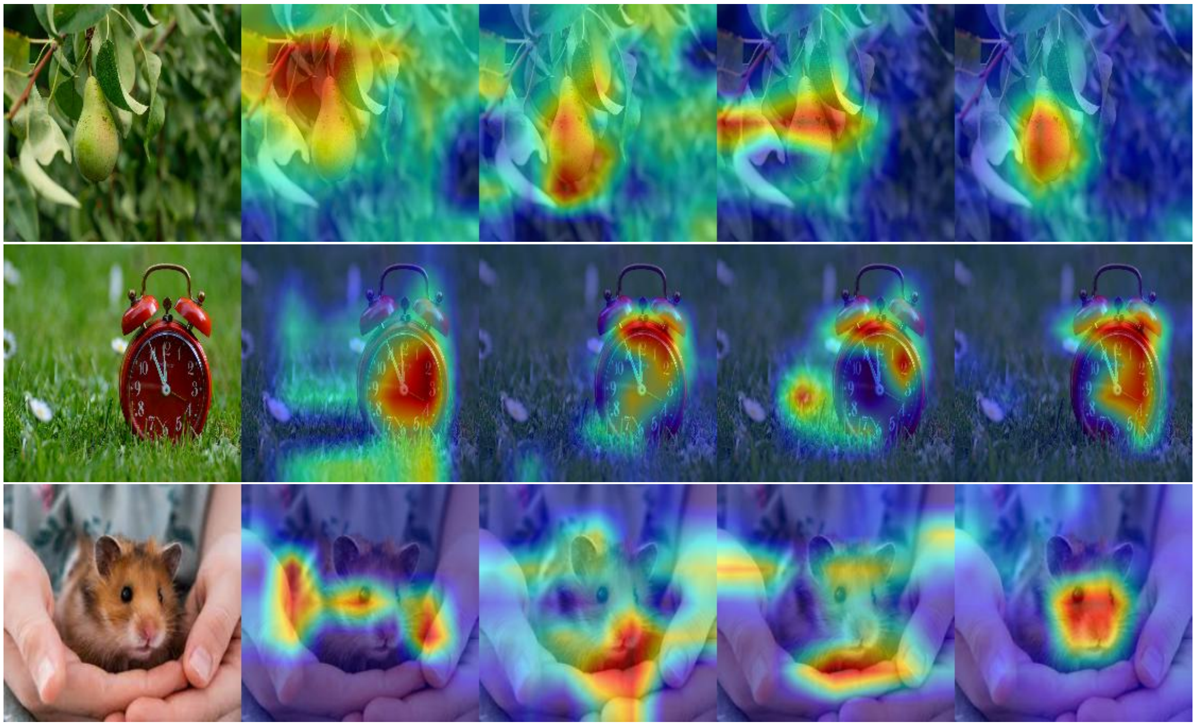

Figure 3. **Attention maps:** We compare the attention maps extracted by the final layer of the architecture. From left to right: original image, attention maps extracted from SwiftFormer [25], EfficientFormerV2 [18], EfficientFormerV2 [18] with element-wise attention, EleViT.

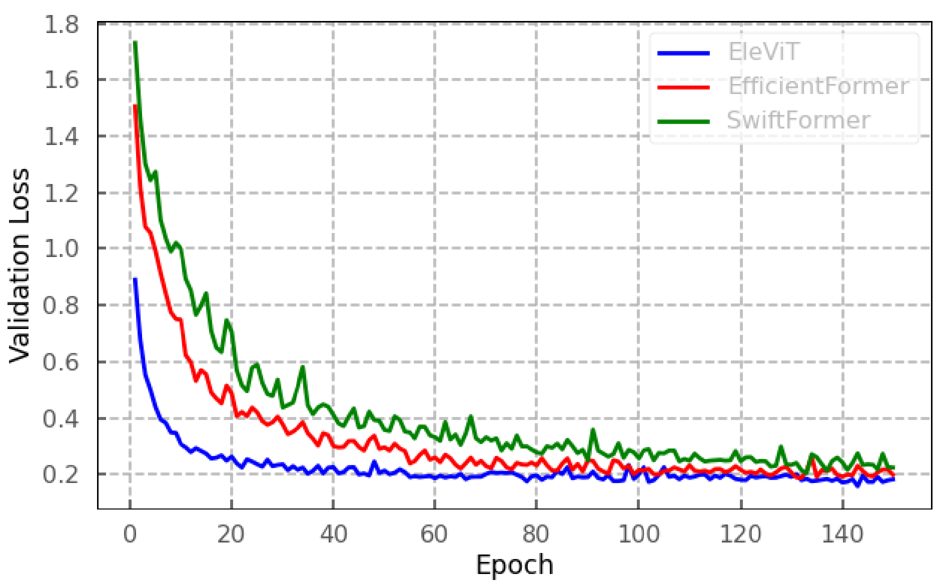

Figure 4. **Validation loss:** for CIFAR10 we compare EleViT to SwiftFormer [25] and EfficientFormerV2 [18]. Our architecture needs less number of epochs to reach the minimum loss.