

MixA: A Mixed Attention approach with Stable Lightweight Linear Attention to enhance Efficiency of Vision Transformers at the Edge

Sabbir Ahmed^{1*} Jingtao Li² Weiming Zhuang² Chen Chen² Lingjuan Lyu^{2†}
¹ Binghamton University, ² Sony AI

Abstract

Vision transformers (ViTs) have become widely popular due to their strong performance across various computer vision tasks. However, deploying ViTs on edge devices remains a persistent challenge due to their high computational demands primarily caused by the over use of self-attention layers with quadratic complexity together with the resource-intensive softmax operation. To resolve this challenge, linear self-attention approach has emerged as an efficient alternative. Nonetheless, current linear attention methods experience considerable performance degradation compared to the softmax-based quadratic attention. Hence, we propose MixA, a novel mixed attention approach that enhances efficiency of ViT models while maintaining comparable performance to softmax-based quadratic attention. MixA takes a pretrained ViT model and analyzes the significance of each attention layer, and selectively apply ReLU-based quadratic attention in the critical layers to ensure high model performance. To enhance efficiency, MixA selects the less critical layers and replaces them with our novel ReLU-based linear attention module called Stable Lightweight Linear Attention (SteLLA). SteLLA utilizes theoretically motivated normalization terms that improve stability of prior ReLU-based linear attention, resulting in better performance (see Figure 1) while achieving significant speedup compared to softmax based quadratic attention (see Figure 2). Experiments conducted on three benchmark vision tasks show that MixA can significantly improve efficiency of ViT models with competitive performance. Notably, MixA can improve inference speed of DeiT-T model by 22% on Apple M1 chip with only $\sim 0.1\%$ accuracy loss.

1. Introduction

Vision Transformers (ViTs) have gained significant popularity and demonstrated state-of-the-art performance across a variety of computer vision tasks, including image classification [8, 28], object detection [3, 4] and semantic seg-

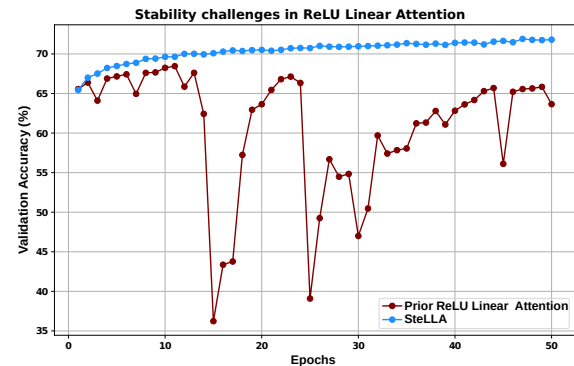


Figure 1. ReLU-based linear attention faces stability challenges due to the prior normalization technique [2, 23], which normalizes each row of the attention matrix by its row sum. In contrast, our proposed SteLLA utilizes theoretically motivated normalization terms to prevent variance explosion, resulting in more stable and superior performance.

mentation [4, 38]. Despite their strong performance, ViTs come with a substantial computational demand, often resulting in slower runtimes compared to competitive convolutional neural networks (CNNs). For example, [20] shows that DeiT-T [28] is more than $10\times$ slower compared to MobileNetV2 [25] on edge inference. Hence, deployment of ViTs remains a significant challenge in resource-constrained edge devices, where inference speed is a critical concern [20, 22].

A key factor limiting the inference speed of ViTs is their self-attention mechanism [2, 10, 15]. In particular, the current self-attention mechanism in ViTs introduces two major issues. First, it results in quadratic computational complexity, $\mathcal{O}(N^2)$, with respect to the sequence length N in all attention layer. Second, the self-attention mechanism heavily utilizes Softmax operations, which involve a large number of exponential function calls during inference. These operations are especially resource-intensive on edge devices [15, 27, 30], where exponential functions require significantly more resources than the basic arithmetic operations, such as multiplication or addition [15].

Previous studies have attempted to address the challenge of heavy computational burden associated with self-

*Work conducted during an internship at Sony AI.

†Corresponding to: lingjuan.lv@sony.com

attention by limiting the global receptive field to a smaller region, such as designing sparse global attention patterns [31, 34] or applying smaller attention windows [11, 17]. While these strategies are effective, they still rely on expensive Softmax operation and often compromise the model’s ability to capture critical information from other regions.

Recently, linear attention designs have emerged as a viable alternative that reduces computational burden of ViTs by decreasing overall complexity. Existing linear attention approaches leverages either complex activation functions [13] or specially designed mapping functions [10, 19] and apply them to queries and keys separately. This separation enables a computational reordering from (query·key)·value to query·(key·value) in self-attention computation, effectively reducing the complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. However, despite this efficiency aspect, current works that solely rely on using linear attention methods either exhibit less expressiveness compared to Softmax-based quadratic attention or introduce additional computational overhead due to the complexity of the adopted kernel functions [19], which limits their appeal for real-world application.

In this paper, we address the limitations of current attention mechanisms by introducing a novel mixed attention approach called MixA. Our proposed MixA takes a pretrained ViT model and modifies its attention mechanism to improve efficiency while ensuring high performance. In particular, MixA consists of two key components. The first component, *Targeted Quadratic Attention*, takes the pretrained ViT model and analyzes the significance of each attention layer for model performance based on a novel importance score. It applies ReLU activation based quadratic attention in the critical layers to ensure high expressiveness. To enhance efficiency, MixA selects the less critical layers and modify them with our second novel component called *Stable Lightweight Linear Attention* (SteLLA). SteLLA applies light ReLU activation function to queries and keys separately enabling the computation reordering from (query·key)·value to query·(key·value) to improve computational complexity and achieve significant speedup over Softmax-based quadratic attention (see Figure 2). However, applying normalization in ReLU based linear attention to mimic Softmax like behavior (i.e., ensuring each row of Attention matrix sums to one) as used in prior works [2, 23] leads to stability challenges and inferior performance. To resolve these challenges, we propose a theoretically motivated normalization technique that ensures stability and improves performance (see Figure 1). Overall, our contributions can be summarized as follows:

- We introduce a novel mixed attention approach, namely MixA, that strategically applies different attention mechanisms. Performance-critical layers utilize ReLU-based

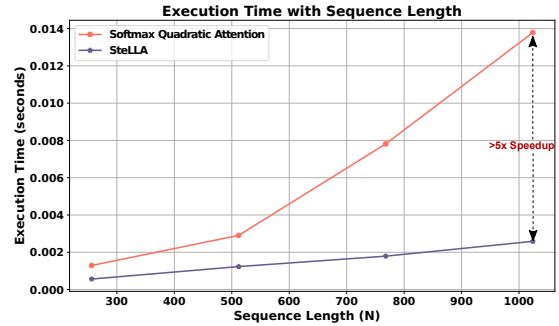


Figure 2. Execution time comparison between Softmax attention (quadratic complexity) and SteLLA (linear complexity) for a single attention layer. SteLLA achieves significant speedup over Softmax attention ($> 5\times$ at sequence length $N=1024$) through its linear complexity and elimination of the computationally expensive Softmax operation. Execution time is measured on Apple M1 chip.

quadratic attention, while less critical layers employ ReLU-based linear attention, resulting in a balanced trade-off between efficiency and performance.

- To determine which layers are performance-critical, we propose a novel importance scoring mechanism that assesses the significance of each attention layer.
- To improve overall efficiency of proposed MixA, we propose SteLLA, a novel ReLU-based linear attention module with theoretically motivated normalization terms that successfully resolves stability challenges in prior ReLU linear attention (see Figure 1) with improved performance and significant speedup over Softmax quadratic attention (see Figure 2).
- We empirically validate the effectiveness of our proposed MixA on image classification, object detection and semantic segmentation tasks using two popular ViT models. The results demonstrate significant improvement in terms of execution time over Softmax-based counterpart while ensuring comparable performance.

2. Related Work

2.1. Vision Transformers

Transformers [29] were initially developed for natural language processing (NLP) tasks, where they quickly became dominant due to their attention mechanism which effectively captures long-range dependencies. Later, Vision Transformers (ViTs) [8] have shown that applying pure transformer architectures directly to sequences of image patches can yield impressive results on various image classification benchmarks, especially with large-scale pretraining. This success has led to a surge in transformer-based models for computer vision. For instance, DeiT [28] introduced advanced training techniques and a novel distillation approach to reduce ViT’s dependence on large datasets, surpassing the original ViT in performance. Models like

PVT [31] and Swin Transformer [17] later integrated pyramid structures to produce multi-scale feature maps, making transformers a versatile backbone for diverse vision tasks. Building on this progress, numerous vision transformers have emerged [9, 33], delivering remarkable results across various benchmarks. Despite their success, ViTs suffer from significant latency that makes them unsuitable for real-world applications, particularly in edge devices.

2.2. Efficient Attention Design

A prime contributor to this high execution time of ViTs is their existing design of self-attention mechanism. To mitigate this, several research efforts have been made targeting efficient attention design. One of the popular approaches involves gradually lowering feature resolution and employing specialized attention patterns to limit the number of tokens being attended to. For example, PVT [31] employs a sparse attention mechanism, selecting tokens for attention from a global perspective. Following this direction, DAT [34] introduces a deformable attention module to create a data-dependent attention pattern. Meanwhile, Swin Transformer [17] takes a different approach by selecting tokens locally, dividing the input into isolated windows. Similarly, NAT [11] adopts a query-centric approach, designing independent tokens for each query. Despite these advancements, many of these approaches still rely on expensive Softmax operator and often compromise the model’s ability to capture critical information from other regions.

In addition to the previously discussed methods, another promising research direction focuses on reducing computational complexity through linear attention mechanisms [5, 13, 15, 35]. Linear attention addresses the inefficiencies of traditional self-attention by replacing the Softmax function with kernel functions, which removes the need to compute the pairwise similarity between queries and keys, \mathbf{QK}^T . As shown in Figure 4, by leveraging the associative property of matrix multiplication, linear attention can reorder computations by first calculating $\mathbf{K}^T \mathbf{V}$, reducing the complexity from $O(N^2d)$ to $O(Nd^2)$.

Although this approach offers computational benefits, designing a linear attention module that matches the effectiveness of Softmax-based attention has remained a persistent challenge. Performer [5] attempts to approximate the Softmax operation using orthogonal random features, while Efficient Attention [13] applies the Softmax function to both queries and keys individually, ensuring that each row of \mathbf{QK}^T sums to one. Other methods such as Nystromformer [35] and SOFT [19] approximate the full self-attention matrix using matrix decomposition techniques. Hydra Attention [1], however, replaces Softmax with cosine similarity and introduces the hydra trick, which further reduces complexity to $O(Nd)$.

Nevertheless, current linear attention designs either lack

the expressive power needed to match Softmax attention or introduce additional computational overhead due to the complexity of the kernel functions used. In this work, we address the limitations of these efficient attention design and propose a novel approach called MixA which employs dual attention based strategy while completely eliminating the need for costly Softmax operation.

3. Preliminaries and Notation

Consider an input of N tokens, where the queries, keys, and values are represented by matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times d}$. The self-attention mechanism within each head of the Multi-Head Self-Attention block is expressed as:

$$\mathbf{O} = \mathbf{A} \cdot \mathbf{V}, \quad (1)$$

where \mathbf{A} denotes the attention matrix, calculated by measuring the similarity between the queries and keys as follows:

$$\mathbf{A} = \text{Softmax} \left(\frac{\mathbf{QK}^T}{\sqrt{d}} \right), \quad (2)$$

where d represents the head dimension.

This self-attention mechanism in (2) computes the similarity between all pairs of queries and keys resulting in a computational complexity of $O(N^2d)$. The number of tokens N is higher than the head dimension d . As a result, applying global self-attention becomes computationally expensive.

4. Inefficiency of Attention Mechanism

Before introducing the details of our proposed method, in this section, we first demonstrate the problem that our work aims to address. Here, we empirically demonstrate the execution time bottlenecks in existing self attention mechanisms.

Observations. The experimental results of our timing analysis are illustrated in Table 1 and Figure 3, which show the breakdown of computation times for different components of ViT model. This leads to our first observation:

Observation I. *The attention layer consumes the majority of the inference time in ViT models, making it a critical bottleneck for efficient edge inference.*

Table 1. Comparison of cumulative execution time measurements between Attention and MLP layers. Measurements are taken on the DeiT-S backbone with a 448x448 input resolution, tested on an Apple M1 chip.

| Layer | Time (ms) | Percentage | Parameters per layer |
|-----------|-----------|------------|----------------------|
| Attention | 121.7 | 75.37% | 0.59M |
| MLP | 39.8 | 24.63% | 1.18M (2×) |

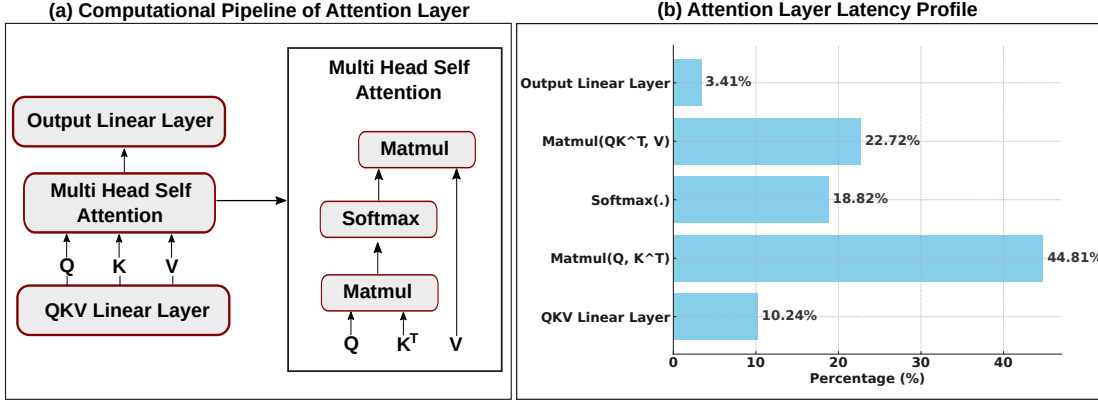


Figure 3. (a) Computational pipeline of a self-attention layer. (b) Latency profile of the self-attention layer, illustrating the percentage of time allocated to each operation within the attention layers. Latency measurements are taken using the DeiT-S backbone with a 448x448 input resolution, tested on an Apple M1 chip.

As shown in Table 1, although attention layers have only half the parameters of MLP layers, they consume more than $3\times$ the computational time. This significant disparity highlights the need for optimizing attention operations for edge devices with limited computational capabilities.

To identify the sources of this computational overhead, we performed a detailed profiling of various operations within the attention mechanism, as illustrated in Figure 3. This profiling reveals our second observation:

Observation II. *Within the attention layer, the quadratic attention operation along with the Softmax operation consume a disproportionate amount of time compared to other operations.*

The QK^T matrix multiplication alone accounts for 44.81% of the total attention time, while the Softmax operation takes up 18.82% of total time even though Softmax contributes to $< 0.5\%$ FLOPs in attention layer. Together, these two operations constitute more than 63% of the attention layer’s computation time. In contrast, the QKV linear transformation and *Output* linear projection operations consume only 10.24% and 3.41% of the total time in the attention layer, respectively.

These observations confirm the latency bottleneck in current attention mechanisms, primarily driven by the quadratic complexity of attention and the computational demands of Softmax operations.

5. Proposed MixA Method

To address the inefficiencies of attention mechanisms in Vision Transformers (ViTs), we introduce a novel approach called MixA, comprising two key components. The first component is called *Targeted Quadratic Attention*, which takes a pretrained Vision Transformer and evaluates the significance of each attention layer through a novel importance score. The importance score quantifies the contribution of each attention layer to model performance. Based on this

score, we selectively modify the Softmax-based quadratic attention in critical layers with efficient ReLU activation based quadratic attention [32]. This modification in critical layers ensures that the overhead associated with Softmax function is eliminated while having minimal impact to model performance. To further enhance efficiency, we modify the attention of less significant layers using our second component called *Stable Lightweight Linear Attention* (SteLLA). SteLLA applies pointwise ReLU activation function independently to keys and queries, enabling a reordering of matrix multiplication from $(QK^T)V$ to $Q(K^T V)$. Since N is often much larger than d (i.e. d is 64 in DeiT, N is 1024 for image resolution of 448×448 with a patch size of 14), this reordering reduces the computational complexity of these layers from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$, as illustrated in Figure 4.

5.1. Targeted Quadratic Attention

The first component of our proposed MixA selects critical attention layers and modifies them with an efficient ReLU activation-based quadratic attention mechanism. To quantify the significance of each attention layer, we introduce an importance score that leverages two critical properties: (1) the impact of attention modifications on the model performance, and (2) the sensitivity of the attention scores to the modification.

To compute the importance score, we begin by performing a forward pass of the input images x through the pretrained ViT model, recording the original attention scores A and the corresponding loss \mathcal{L} . Next, we perform backpropagation to calculate the gradient of the loss with respect to each attention head, denoted as $\nabla_{A_i} \mathcal{L}$. The gradient for the i -th attention head indicates how sensitive the loss function is to small changes in A_i , capturing the impact of attention modifications on model performance.

Next, we modify the attention layer to replace the Softmax-based quadratic attention layers with linear atten-

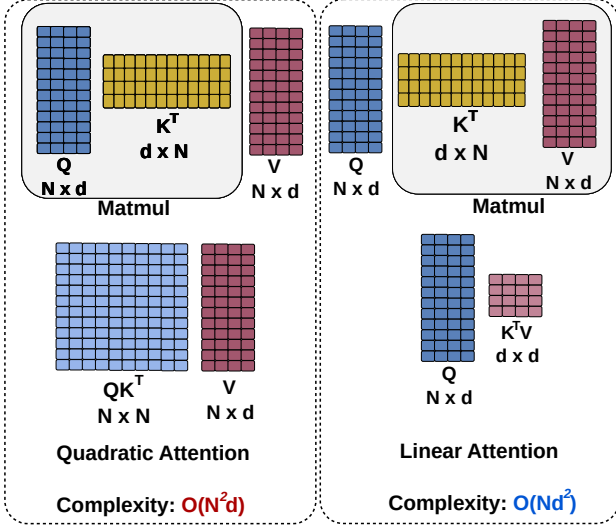


Figure 4. Comparison of attention mechanisms with quadratic and linear computational complexity. $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times d}$ denote the query, key, and value matrices, respectively. Traditional Softmax-based attention in (2) requires pairwise similarity computation between queries and keys, resulting in a complexity of $\mathcal{O}(N^2d)$. Linear attention, on the other hand, applies activation function or mapping functions to \mathbf{Q} and \mathbf{K} separately enabling calculation of $\mathbf{K}^T \mathbf{V}$ first, reducing complexity to $\mathcal{O}(Nd^2)$. N is higher than d in ViTs [17, 28].

tion in (7) and perform another forward pass using the same input \mathbf{x} to obtain the modified attention scores \mathbf{A}' and compute the perturbation in the attention scores for each head as $\Delta \mathbf{A}_i = \mathbf{A}'_i - \mathbf{A}_i$. This perturbation reflects the sensitivity of the attention layer to the applied modification.

The proposed importance score \mathcal{I}_A is then calculated as:

$$\mathcal{I}_A = \frac{1}{H} \sum_{i=1}^H \left| \text{Tr} \left((\nabla_{\mathbf{A}_i} \mathcal{L})^\top \Delta \mathbf{A}_i \right) \right|, \quad (3)$$

where H denotes the total number of attention heads in the layer and $\text{Tr}(\cdot)$ denotes the trace operator. By taking the absolute value and averaging across all heads, we obtain a scalar importance score that reflects the overall significance of the attention layer. A higher value of the importance score \mathcal{I}_A indicates that the corresponding attention layer is critical for maintaining the model's performance.

After computing the importance scores for each attention layer, we select the layers with the highest importance scores for quadratic attention placement. To formalize, let $\mathcal{L}_A = \{l_1, l_2, \dots, l_L\}$ denote the set of all attention layers in the model. We select a subset of layers $\mathcal{S} \subset \mathcal{L}_A$ as follows

$$\mathcal{S} = \{l_i \mid l_i \in \mathcal{L}_A \text{ and } \mathcal{I}_A^{l_i} \in \text{top-}k(\mathcal{I}_A)\}$$

where k is a hyperparameter representing the number of layers chosen for quadratic attention.

Even though placing quadratic attention in some layers incurs computational cost, we ensure that the quadratic attention is Softmax-free. To ensure this, we utilize ReLU as activation function for computing this quadratic self attention which is formulated as follows

$$\mathbf{O} = \frac{1}{\alpha_Q} \cdot \mathbf{A}_Q \cdot \mathbf{V}, \quad (4)$$

where the ReLU-based quadratic attention matrix \mathbf{A}_Q in (4) is calculated as:

$$\mathbf{A}_Q = \text{ReLU} \left(\frac{\mathbf{Q} \mathbf{K}^T}{\beta_Q} \right), \quad (5)$$

where β_Q in (5) and α_Q in (4) are normalization terms following Theorem 5.1 and Theorem 5.2.

Theorem 5.1. Consider the query and key matrices $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{N \times d}$ and assume that the components of query $\mathbf{Q}_i \in \mathbb{R}^d$ and key $\mathbf{K}_j \in \mathbb{R}^d$ are independent random variables following standard normal distribution. Then the variance of dot-product between \mathbf{Q}_i and \mathbf{K}_j follows

$$\text{Var}(\mathbf{Q}_i^T \mathbf{K}_j) \in \mathcal{O}(d)$$

where d is the dimension of the attention head.

Here, the independent assumption is followed by [29] and the proof is provided in the supplementary material. According to Theorem 5.1, we propose a scaling factor $\beta_Q = \beta'_Q \sqrt{d}$ in dot-product attention computation in (5) to counteract the variance explosion, where β'_Q is a learnable parameter initialized as 1.

Theorem 5.2. Consider an attention matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ and value matrix $\mathbf{V} \in \mathbb{R}^{N \times d}$. Assume that the variance of product of elements A_{ij} and V_{jk} is bounded, i.e., $\text{Var}(\mathbf{A}_{ij} \mathbf{V}_{jk}) \leq C$ for some constant C . Then the variance of elements of the resultant matrix, i.e., $\mathbf{O}_{ik} = \sum_{j=1}^N \mathbf{A}_{ij} \mathbf{V}_{jk}$ has the following growth rate

$$\text{Var}(\mathbf{O}_{ik}) \in \mathcal{O}(N^2)$$

where N is the number of tokens in the sequence.

The proof of Theorem 5.2 is provided in the supplementary material. According to Theorem 5.2, we use a scaling factor $\alpha_Q = \alpha'_Q N$ in (4) to counteract the variance explosion when Attention matrix \mathbf{A}_Q is multiplied by the Values matrix \mathbf{V} , and α'_Q is a learnable parameter initialized as 1.

Overall, by leveraging the proposed importance score, we effectively identify and select the most critical layers for ReLU-based quadratic attention and avoid the overhead associated with the Softmax operation. However, applying quadratic attention to less critical layers is unnecessary. Therefore, to enhance overall efficiency, we propose to utilize linear attention mechanism in these less critical layers.

Table 2. Performance comparison of different ViT models [17, 28] for classification task on ImageNet-1K dataset [7] with Softmax-based quadratic attention [29] and proposed MixA.

| Method | #Params | FLOPs | Accuracy (%) | Execution Time (ms) | | | |
|-------------|---------|--------|--------------|---------------------|------------------|--------------|------------------|
| | | | | Apple M1 | Speedup Gain (%) | Raspberry Pi | Speedup Gain (%) |
| DeiT-T [28] | 5.7M | 3.40G | 74.66 | 12.57 | - | 346.76 | - |
| MixA-DeiT-T | 5.7M | 3.17G | 74.54 | 10.30 | 22.04 | 299.90 | 15.63 |
| DeiT-S [28] | 22M | 12.25G | 80.76 | 31.00 | - | 863.38 | - |
| MixA-DeiT-S | 22M | 11.79G | 80.59 | 27.07 | 14.52 | 778.07 | 10.96 |
| Swin-T [17] | 29M | 9.79G | 81.76 | 49.74 | - | 940.46 | - |
| MixA-Swin-T | 29M | 9.24G | 81.28 | 40.66 | 22.33 | 819.97 | 14.69 |
| Swin-S [17] | 51M | 18.83G | 83.59 | 79.06 | - | 1674.12 | - |
| MixA-Swin-S | 51M | 17.89G | 82.88 | 61.67 | 28.20 | 1408.70 | 18.84 |

5.2. Stable Lightweight Linear Attention

The second component of our proposed MixA is a novel lightweight attention mechanism with linear computational complexity. Here, we apply lightweight ReLU activation function to queries and keys separately enabling the computation reordering from $(\mathbf{Q}\mathbf{K}^T)\mathbf{V}$ to $\mathbf{Q}(\mathbf{K}^T\mathbf{V})$ in the less critical layers to reduce their computational complexity from quadratic to linear (see Figure 4). However, applying normalization in ReLU-based linear attention to mimic Softmax-like behavior (i.e., ensuring each row of Attention matrix sums to one) as used in prior works [2, 23] leads to instability and degraded performance (see Figure 1). To resolve this, we propose a theoretically motivated normalization technique that ensures stability and improved performance which we formulate as follows

$$\mathbf{O} = \frac{1}{\alpha_L} \cdot \mathbf{A}_L \cdot \mathbf{V}, \quad (6)$$

where the linear attention matrix \mathbf{A}_L is calculated as:

$$\mathbf{A}_L = \frac{\text{ReLU}(\mathbf{Q})\text{ReLU}(\mathbf{K})^T}{\beta_L} \quad (7)$$

where α_L in (6) and β_L in (7) are our proposed normalization terms derived in Theorem 5.2 and Theorem 5.3.

Theorem 5.3. Consider the query and key matrices $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{N \times d}$ and assume that the components of query $\mathbf{Q}_i \in \mathbb{R}^d$ and key $\mathbf{K}_j \in \mathbb{R}^d$ are independent random variables following standard normal distribution. Then the variance of dot-product between $\text{ReLU}(\mathbf{Q}_i)$ and $\text{ReLU}(\mathbf{K}_j)$ vectors follows

$$\text{Var}(\text{ReLU}(\mathbf{Q}_i)^T \text{ReLU}(\mathbf{K}_j)) \in \mathcal{O}(d)$$

where d is the dimension of the attention head.

Here, the independent assumption is followed by [29] and the proof is provided in the supplementary material. According to Theorem 5.3, we propose a scaling factor $\beta_L = \beta'_L \sqrt{d}$ in (7) to counteract the variance explosion, where β'_L is a learnable parameter initialized as 1. And

according to Theorem 5.2, we propose a scaling factor $\alpha_L = \alpha'_L N$ in (6) to mitigate variance explosion when attention matrix \mathbf{A}_L is multiplied by the values matrix \mathbf{V} , where α'_L is also a learnable parameter initialized as 1.

6. Experimental Results

In this section, we evaluate the performance of proposed MixA on image classification, object detection and semantic segmentation tasks using two popular ViT architectures: DeiT [28] and Swin [17]. We consider primarily “small” and “tiny” variants as we focus on accelerating edge inference.

6.1. Datasets and Implementation Details

Datasets. We evaluate our approach on several benchmark datasets across multiple tasks. For image classification, we use the widely adopted ImageNet-1K dataset [7], which includes 1.28 million training images and 50,000 validation images spanning 1,000 classes. For object detection, we utilize the COCO dataset [16], consisting of 118,000 training images and 5,000 validation images. For semantic segmentation, we conduct experiments on the ADE20K dataset [40] with 20,000 training images and 2,000 validation images.

Experimental Settings and Execution Times. For classification task, we take pretrained ViT models [17, 28] and fine-tune them after applying MixA. For finetuning the models, we utilize the cross-entropy loss and standard knowledge distillation loss [12], using pretrained models as teachers. This fine-tuning process is conducted over 150 epochs with the AdamW optimizer [18] and a cosine learning rate schedule, including 10 warm-up epochs with a base learning rate of 1×10^{-4} . To avoid overfitting, we apply the augmentations used in DeiT [28]. In addition, a weight decay of 0.05 is also used. For fair comparison, we carry out exact fine-tuning process and finetune the pretrained ViT models with Softmax-based quadratic attention under the same settings and report their results. Similarly, we carry out the same fine-tuning process to report performance of existing linear attention mechanisms [13, 15, 23].

For object detection and semantic segmentation tasks, we use ViT backbones with their respective classification

Table 3. Performance comparison of different ViT backbones [17, 28] for semantic segmentation tasks on ADE20k dataset [40] with Softmax-based quadratic attention [29] and proposed MixA.

| Method | #Params | FLOPs | mIoU (%) | Execution Time (ms) | | | |
|---------------------|---------|--------|----------|---------------------|------------------|--------------|------------------|
| | | | | Apple M1 | Speedup Gain (%) | Raspberry Pi | Speedup Gain (%) |
| DeiT-Adapter-T [4] | 12M | 29.12G | 34.7 | 211.54 | - | 3798.57 | - |
| MixA-DeiT-Adapter-T | 12M | 26.48G | 33.8 | 187.67 | 12.72 | 3539.81 | 7.31 |
| Swin-T [17] | 32M | 39.17G | 35.3 | 148.93 | - | 3299.54 | - |
| MixA-Swin-T | 32M | 36.97G | 35.8 | 115.70 | 28.72 | 2935.70 | 12.39 |

Table 4. Performance comparison of different ViT backbones [17, 28] for object detection task on COCO dataset [7] with Softmax-based quadratic attention [29] and proposed MixA.

| Method | #Params | FLOPs | mAP (%) | Execution Time (ms) | | | |
|---------------------|---------|--------|---------|---------------------|------------------|--------------|------------------|
| | | | | Apple M1 | Speedup Gain (%) | Raspberry Pi | Speedup Gain (%) |
| DeiT-Adapter-T [4] | 14M | 29.12G | 33.9 | 208.62 | - | 3807.66 | - |
| MixA-DeiT-Adapter-T | 14M | 26.48G | 33.5 | 185.23 | 12.62 | 3514.72 | 8.33 |
| Swin-T [17] | 34M | 39.17G | 34.6 | 146.49 | - | 3315.01 | - |
| MixA-Swin-T | 34M | 36.97G | 32.6 | 112.22 | 30.53 | 2945.79 | 12.53 |

checkpoints, integrating them with Faster-RCNN [24] and SemanticFPN [14] models. For object detection, we train the object detection model for 36 epochs, and for semantic segmentation, we train for 40k iterations and use a batch size of 16 across both tasks. For additional implementation details and more ablation studies (e.g., effect of different k), we direct the reader to the Supplementary section.

We use Raspberry Pi 4 Model B and MacBook Pro with Apple M1 Max chip to report the execution times. Following the standard approach in [15, 20], we measure execution time on these devices by first performing five warm-up runs. We then record the execution time across 100 inference runs and report the average execution time. For object detection and semantic segmentation tasks, we report the execution times and FLOPs of the backbone networks for direct comparison between different ViT backbones.

6.2. Main Results

Image Classification. Table 2 presents performance comparison between Softmax-based quadratic attention and our proposed MixA on the ImageNet-1K dataset. As shown in the table, with MixA, each model variant shows a reduction in FLOPs and execution times on both Apple M1 and Raspberry Pi devices, while maintaining similar accuracy levels. Notably, MixA achieves over 20% speedup in DeiT-T and Swin-T, with only a minor accuracy drop of approximately 0.1% and 0.5%, respectively. The most significant gains are seen in Swin-S, where MixA improves execution speed by up to 28.20% on Apple M1 and 18.84% on Raspberry Pi, demonstrating effectiveness of proposed approach for improving execution time.

Semantic Segmentation. Table 3 presents performance comparison between Softmax-based quadratic attention and MixA-based backbones for DeiT-Adapter-T and Swin-T, integrated with the Semantic FPN model, on the ADE20K

dataset. Similar to object detection, we adopt an edge friendly resolution of 448×448 to report the model performance. And similar to object detection, MixA provides comparable mean Intersection over Union (mIoU) values in semantic segmentation while reducing FLOPs and achieving significant execution time speedups of ViT backbones. For example, MixA-DeiT-Adapter-T reduces FLOPs from 29.12G to 26.48G, with a minor mIoU degradation $< 1\%$, and achieves a speedup of 12.39% on Apple M1 and 7.31% on Raspberry Pi. Notably, MixA-Swin-T improves its segmentation performance with an mIoU improvement to 35.8% and reduces FLOPs and gains speedup of over 28% on Apple M1 and 12% on Raspberry Pi.

Object Detection. Table 4 presents performance comparison of Softmax-based quadratic attention and MixA based backbones for DeiT-Adapter-T and Swin-T with the Faster R-CNN model as object detection head on the COCO dataset. We use DeiT-Adapter since DeiT itself does not perform well for dense vision tasks for its lacking vision-specific inductive bias [4]. We use an edge-friendly resolution of 448×448 to evaluate model performance. From the results, we find that, for each model variant, MixA provides a notable reduction in both FLOPs and execution times (12–30% speedup gain) across Apple M1 and Raspberry Pi devices for ViT backbones, while maintaining comparable mean Average Precision (mAP).

Table 5. Comparison with Other Linear Attentions with DeiT-T on ImageNet-1K.

| Linear Attention | #Params | FLOPs | Accuracy (%) |
|---------------------|---------|-------|--------------|
| CosFormer [23] | 5.7M | 2.94G | 69.05 |
| Efficient Attn [26] | 5.7M | 2.94G | 71.80 |
| SimA [15] | 5.7M | 2.94G | 72.73 |
| Stella (Ours) | 5.7M | 2.94G | 73.71 |

6.3. Ablation Study

Comparison with Other Linear Attentions. On the DeiT-T model, we compare our proposed SteLLA module with three established linear attention methods: CosFormer [23], Efficient Attention [26], and SimA [15]. As shown in Table 5, SteLLA outperforms the prior linear attention approaches. Notably, our normalization technique enables SteLLA to achieve a 4.66% improvement over CosFormer [23], which uses similar ReLU-based linear attention but with a different normalization approach (normalizes each row of Attention matrix by corresponding row sum). This improved performance highlights the effectiveness of the proposed normalization technique in SteLLA.

Effect of different components of MixA. Next, we assess the impact of each component of our proposed MixA by progressively incorporating them. Starting from the proposed SteLLA, we observe that adding targeted quadratic attention results in an accuracy improvement of +0.83%, as shown in Table 6.

Table 6. Effect of different components of MixA.

| Method | #Params | FLOPs | Accuracy (%) |
|--------------------------|---------|-------|--------------|
| SteLLA | 5.7M | 2.94G | 73.71 |
| SteLLA + TQA ($k = 6$) | 5.7M | 3.17G | 74.54 |

Impact of Targeted Quadratic Attention. Table 7 compares the effect of targeted quadratic attention and applying quadratic attention to randomly chosen layers. As shown in Table 7, placing quadratic attention on random layers achieves an accuracy of 73.98%. In contrast, applying targeted quadratic attention boosts accuracy to 74.54% without increasing FLOPs. This demonstrates that selecting layers based on their importance score significantly enhances model performance, confirming the effectiveness of importance score-based layer selection.

Table 7. Impact of Targeted Quadratic Attention on MixA.

| Method | #Params | FLOPs | Accuracy (%) |
|--------------------------|---------|-------|--------------|
| SteLLA + QA ($k = 6$) | 5.7M | 3.17G | 73.98 |
| SteLLA + TQA ($k = 6$) | 5.7M | 3.17G | 74.54 |

Impact of scaling parameters ($\alpha'_L, \beta'_L, \alpha'_Q, \beta'_Q$). Table 8 shows that incorporating learnable scaling parameters in MixA improves accuracy on DeiT-T [28] from 74.44% to 74.54%. This suggests they help prevent variance explosion and enhance the Attention mechanism.

Table 8. Impact of learnable scaling parameters ($\alpha'_L, \beta'_L, \alpha'_Q, \beta'_Q$) on proposed MixA evaluated on DeiT-T.

| Method | #Params | FLOPs | Accuracy (%) |
|----------------------------|---------|-------|--------------|
| MixA (wo learnable params) | 5.7M | 3.17G | 74.44 |
| MixA | 5.7M | 3.17G | 74.54 |

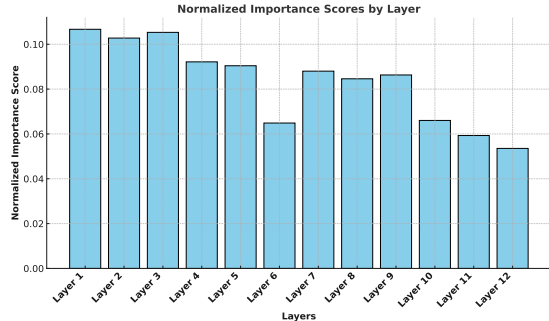


Figure 5. Normalized importance scores of DeiT-T.

Impact of different k . Table 9 illustrates the effect of varying k on MixA’s performance using DeiT-T. While the parameter count remains constant at 5.7M, FLOPs increase from 3.05G to 3.28G as k grows from 3 to 9. Accuracy peaks at 74.54% for $k = 6$ but shows no further gains at $k = 9$, indicating that $k = 6$ balances computational cost and performance, justifying its selection.

Table 9. Impact of different k on MixA, evaluated on DeiT-T.

| Method | #Params | FLOPs | Accuracy (%) |
|----------------|---------|-------|--------------|
| MixA ($k=3$) | 5.7M | 3.05G | 74.28 |
| MixA ($k=6$) | 5.7M | 3.17G | 74.54 |
| MixA ($k=9$) | 5.7M | 3.28G | 74.54 |

Layerwise Importance Score. Figure 5 visualizes the distribution of importance scores across the layers of DeiT-T. The distribution of scores indicate that the attention of earlier layers are more sensitive and contribute more significantly to model performance compared to the later layers. This pattern may arise because the earlier layers are primarily responsible for local feature extraction, making them better suited to benefit from the enhanced capacity of quadratic attention. In contrast, the later layers, which focus on capturing global context, may perform adequately with linear attention.

7. Conclusion

In this work, we propose MixA, a novel and efficient attention approach that enhances the efficiency of Vision Transformer models while achieving performance comparable to traditional, resource-intensive Softmax-based quadratic attention. MixA uses a novel importance score to apply ReLU-based quadratic attention in critical layers for high expressiveness, while improves efficiency by incorporating a new linear attention module, SteLLA, in less critical layers. Extensive experiments on tasks including image classification, object detection, and semantic segmentation demonstrate that MixA can be seamlessly integrated into Vision Transformers, achieving an optimal balance between efficiency and model performance.

References

- [1] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. Hydra attention: Efficient attention with many heads. In *European Conference on Computer Vision*, pages 35–49. Springer, 2022. 3
- [2] Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17302–17313, 2023. 1, 2, 6
- [3] 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. 1
- [4] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022. 1, 7, 2
- [5] Krzysztof Choromanski, Valerii Likhoshesterov, 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
- [6] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 2
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6, 7
- [8] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2
- [9] 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. 3
- [10] Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5961–5971, 2023. 1, 2
- [11] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6185–6194, 2023. 2, 3
- [12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Proceedings of the NIPS Deep Learning and Representation Learning Workshop*, 2015. 6, 2
- [13] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and Francois Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. *Proceedings of the International Conference on Machine Learning (ICML)*, pages 5156–5165, 2020. 2, 3, 6
- [14] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019. 7
- [15] Soroush Abbasi Koohpayegani and Hamed Pirsiavash. Sima: Simple softmax-free attention for vision transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2607–2617, 2024. 1, 3, 6, 7, 8, 2
- [16] 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 *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6
- [17] 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. 2, 3, 5, 6, 7
- [18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2019. 6, 2
- [19] 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. 2, 3
- [20] Sachin Mehta and Mohammad Rastegari. Mobilevit: lightweight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021. 1, 7
- [21] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2
- [22] Junting Pan, Adrian Bulat, Fuwen Tan, Xiatian Zhu, Lukasz Dudziak, Hongsheng Li, Georgios Tzimiropoulos, and Brais Martinez. Edgevits: Competing light-weight cnns on mobile devices with vision transformers. In *European Conference on Computer Vision*, pages 294–311. Springer, 2022. 1
- [23] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. *arXiv preprint arXiv:2202.08791*, 2022. 1, 2, 6, 7, 8
- [24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 7, 2
- [25] 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. 1
- [26] 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. [7](#), [8](#)
- [27] Jacob R Stevens, Rangharajan Venkatesan, Steve Dai, Brucek Khailany, and Anand Raghunathan. Softmax: Hardware/software co-design of an efficient softmax for transformers. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 469–474. IEEE, 2021. [1](#)
- [28] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. [1](#), [2](#), [5](#), [6](#), [7](#), [8](#)
- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. [2](#), [5](#), [6](#), [7](#)
- [30] Ihor Vasylytsov and Wooseok Chang. Efficient softmax approximation for deep neural networks with attention mechanism. *arXiv preprint arXiv:2111.10770*, 2021. [1](#)
- [31] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. [2](#), [3](#)
- [32] Mitchell Wortsman, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Replacing softmax with relu in vision transformers. *arXiv preprint arXiv:2309.08586*, 2023. [4](#)
- [33] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22–31, 2021. [3](#)
- [34] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4794–4803, 2022. [2](#), [3](#)
- [35] 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*, pages 14138–14148, 2021. [3](#)
- [36] Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. [2](#)
- [37] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. [2](#)
- [38] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. [1](#)
- [39] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13001–13008, 2020. [2](#)
- [40] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019. [6](#), [7](#)