

Resource-efficient Hybrid X-formers for Vision

Pranav Jeevan Amit Sethi
Indian Institute of Technology Bombay
Mumbai, India

{pranavjp@ee., asethi@}iitb.ac.in

Abstract

Although transformers have become the neural architectures of choice for natural language processing, they require orders of magnitude more training data, GPU memory, and computations in order to compete with convolutional neural networks for computer vision. The attention mechanism of transformers scales quadratically with the length of the input sequence, and unrolled images have long sequence lengths. Plus, transformers lack an inductive bias that is appropriate for images. We tested three modifications to vision transformer (ViT) architectures that address these shortcomings. Firstly, we alleviate the quadratic bottleneck by using linear attention mechanisms, called X-formers (such that, $X \in \{\text{Performer}, \text{Linformer}, \text{Nyströmformer}\}$), thereby creating Vision X-formers (ViXs). This resulted in up to a seven times reduction in the GPU memory requirement. We also compared their performance with FNet and multi-layer perceptron mixers, which further reduced the GPU memory requirement. Secondly, we introduced an inductive prior for images by replacing the initial linear embedding layer by convolutional layers in ViX, which significantly increased classification accuracy without increasing the model size. Thirdly, we replaced the learnable 1D position embeddings in ViT with Rotary Position Embedding (RoPE), which increases the classification accuracy for the same model size. We believe that incorporating such changes can democratize transformers by making them accessible to those with limited data and computing resources.

1. Introduction

Transformers have revolutionised the natural language processing (NLP) domain with their ability to handle context over long sequences of data and achieving human-level accuracy for various tasks, such as language translation, text summarization, question answering, language modeling, and text generation [25, 3]. On the other hand, in recent years, vision applications have been almost completely dominated by the convolutional neural network (CNN) ar-

chitectures, which can exploit the two-dimensional (2D) structure of images using inductive priors, such as translational equivariance due to convolutional weight sharing and partial scale invariance due to pooling operations. Even though CNNs possess these advantages in handling image data, they cannot scale up receptive fields without increasing network depth and require several layers, especially pooling-type mechanisms, to capture long range dependencies. While the effective weights are dynamically calculated based on the inputs in attention mechanisms for transformers, popular CNN architectures, such as ResNet [7] and InceptionNet [21] lack an attention mechanism and use static weights for each input, although there are exceptions, such as squeeze and excitation nets [10, 9]. Thus, CNNs can further benefit from carefully crafted attention mechanisms to capture information from all spatial locations of an input. These advantages of attention mechanisms over vanilla CNNs have motivated the research into transformer architectures suitable for vision applications [12].

In spite of their advantages over CNNs, transformers with their default attention mechanisms have found limited use in vision due to their quadratic complexity with respect to sequence length. Images are fed into transformers as long unrolled 1D sequences of pixels or patches. For example, an image of size 256×256 pixels becomes a sequence of length 65,536 pixels. The computational and memory requirements also scale quadratically as we use images of higher resolutions. Transformers also lack inductive priors to exploit the 2D structure of the images. A general architecture without a strong inductive prior makes transformers suitable for applications in multiple domains, provided one has enough data and computational resources. However, the required data, resources, and time for training transformers to achieve accuracy comparable with CNNs on images is not suitable for resource-constrained scenarios [12]. The inductive prior of 2D convolutions in the latter allows them to learn from relatively smaller datasets.

We propose three specific improvements to vision transformers to reduce the resources required to process images: (i) replacing the quadratic attention in ViT with linear atten-

tion mechanisms such as Linformer [26], Performer [2] and Nyströmformer [29] creating the Vision X-former (ViX), (ii) replacing the initial linear embedding layer by convolutional layers in ViX creating a hybrid ViX, and (iii) replacing the popular learnable 1D position embedding with Rotary Position Embedding (RoPE) [20]. While, individually, some of the sub-quadratic attention mechanisms have been tested on images, none of the testing started with images in mind for a thorough comparison of various attention mechanisms for vision. Our idea of using convolutional embedding layers was also developed in parallel by another group [28], while the use of RoPE for vision transformers has not been proposed before. These improvements help us use smaller network sizes for images, which result in the processing of longer sequences at lesser cost. Our experiments suggest that each of these changes independently contribute to the increase in image classification accuracy on relatively smaller datasets using supervised learning. We also demonstrate these improvements across alternative transformer models, such as LeViT, CCT (Compact Convolutional Transformer), CvT (Convolutional vision Transformer) and PiT (Pooling-based Vision Transformer) when we tested them after making the proposed changes to their architectures. We also observed similar trends in the performance of FNet and MLP mixer-based vision transformers, when these are modified as proposed.

2. Related Works

2.1. Vision transformers

Transformers have been applied to unrolled pixel sequences [18]. With Vision Transformer (ViT), it was proposed that applying the original transformer model to a sequence of sub-images (a.k.a. patches) rather than individual pixels of an image shortens the sequence length [4], as shown in Figure 1. Two-dimensional image patches are flattened into a vector and fed to the transformer as a sequence. These vectorized patches are then projected onto a patch embedding using a linear layer and 1D learnable position embedding is attached with it to encode location information. A learnable class embedding is prepended to the sequence of embedded patches. The output representation corresponding to this position is used as the global image representation for the image classification task. ViT achieved excellent results on image classification compared to state-of-the-art CNN models.

However, architecture that use unrolled pixel or even patch sequences have to be pre-trained on a large dataset (e.g., 300 million images [4]) and yet they show poor performance on medium-size datasets due to the lack of inductive priors for images. A larger number of images is required to discover the knowledge rules when a suitable inductive prior, e.g. 2D translational equivariance, is not used.

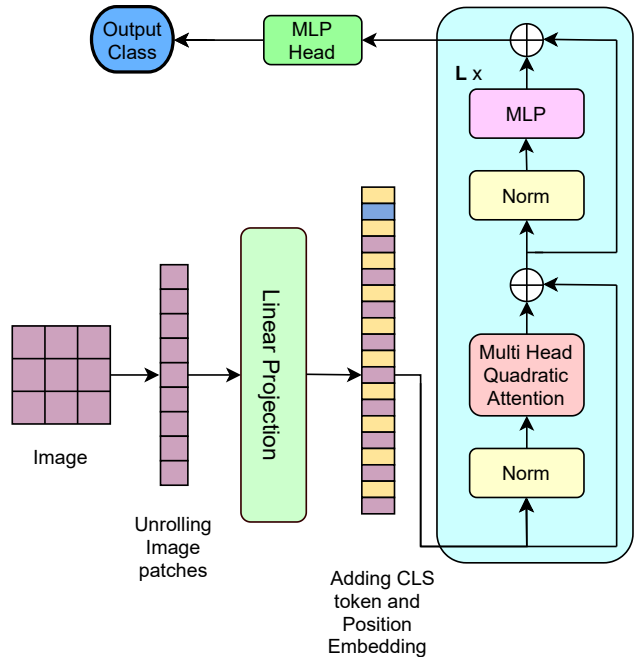


Figure 1. ViT architecture

2.2. Linear transformers

Recently, there has been an influx of proposals for efficient transformers (X-formers) to tackle the problem of quadratic complexity of the number of attention weights with respect to the sequence length [23]. These X-formers, such as Linformer [26], Performer [2], Linear Transformer [11] and Nyströmformer [29] claim linear complexity in computing attention, which makes them ideal candidates for analyzing long pixel sequences encountered in vision tasks. The Linformer projects the length dimension of keys and values to a lower dimensional representation [26]. Performer [2] and Linear transformer [11] use kernels to cleverly re-write the attention mechanism to avoid computing the $N \times N$ attention matrix. Nyströmformer repurposes the Nyström method for approximating self-attention using landmark (or Nyström) points to reconstruct the softmax matrix in self-attention [29]. The Long Range Arena (LRA) benchmark was established to compare the performance and efficiency of these X-formers on various long sequence tasks, including image classification [22]. Most of the applications of linear transformers discussed above have been limited to the NLP domain. We exploit the linear complexity of these attention mechanisms to create a more efficient transformer for vision applications that can handle long sequences and smaller patch sizes with reduced GPU memory.

2.3. Convolutional embedding layers for ViT

It was observed in [1] and [4] that early layers in the transformer learn locally connected patterns which resemble convolutions. This led to the development of many hybrid architectures inspired by transformers and CNNs. LeViT is such a hybrid architecture for fast inference image classification using both convolutional layers and attention [5]. LeViT uses convolutional embedding instead of patch-wise projection and uses 2D relative positional biases instead of initial absolute positional bias used in ViT. It also down-samples the image in stages, uses an extra non-linearity in attention and also replaces layer-norm with batch-norm. CvT improves upon ViT by using a hierarchy of transformers containing a new convolutional token embedding, and a convolutional transformer block [27]. CCT eliminates the requirement for class token and positional embedding through a novel sequence pooling strategy and the use of convolutions [6]. PiT uses the spatial dimension reduction principle of CNN to improve model capability and generalization performance compared to ViT [8]. LeViT, CvT and CCT use convolution layers to down-sample the image size to reduce the sequence length. Additionally, they added multiple convolutional layers between the attention blocks, which increases the number of model parameters since CNNs have more parameters than transformers. Therefore, we used a single convolutional block in our architecture replacing the initial linear embedding layer of ViT to provide the required inductive priors to our model, without down-sampling the image size.

2.4. Alternative positional embedding

Rotary position embedding (RoPE) encodes the absolute positional information with a rotation matrix and incorporates an explicit relative position dependency in the self-attention mechanism [20]. It captures decaying inter-token dependency with increasing relative distances and can be used easily with linear self-attention mechanisms with relative position encoding. The previous works using RoPE have been limited to NLP and molecular domains [19]. We observed the change in classification accuracy when the standard learnable 1D positional embedding is replaced with RoPE in the ViT.

2.5. Alternative mixing models

Recently, new architectures were introduced that replace attention with other mechanisms that enable mixing of information between tokens. FNet replaces the self-attention layer in the transformer with the Fourier Transform that does not need to be learned [16]. FNet was tested mostly in NLP tasks and its performance was compared to language models like BERT. MLP-mixer replaces attention using two types of multi-layer perceptrons (MLPs) which are applied independently first to image patches and then across patches

[24]. In this paper, we compare the performance and GPU consumption of these models in supervised image classification by replacing attention in ViT with these alternative mixing mechanisms. We also observe the improvement in accuracy when we add convolutional layers to create the embedding.

3. Proposed Architectures for ViX

We replace the regular quadratic attention in ViT with linear attention such as Linformer, Performer and Nyströmformer, creating the Vision X-former (ViX), where $X \in \{\text{Performer, Linformer, Nyströmformer}\}$, and compare their accuracy and computational requirements for image classification¹. We set the patch size of the ViT to one so that each pixel is considered as a separate token and the entire image is unrolled to create the full-length image token sequence. This is done to observe the performance in long sequence image classification. We also test the performance of FNet and MLP mixer architectures of similar depth to compare them with ViX. The ViX architecture is shown in Figure 2.

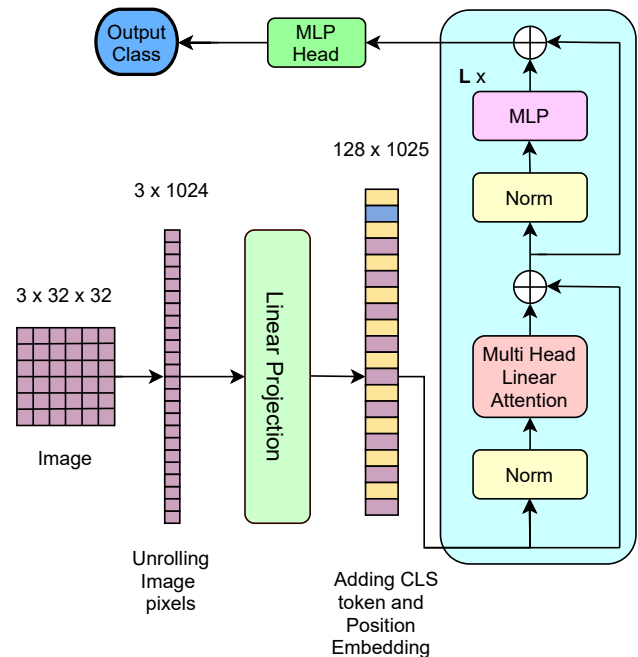


Figure 2. ViX architecture

The lack of inductive priors restricts attention from exploiting the 2D nature and symmetries of the images in the low data regime. We therefore use convolutional layers to replace the initial linear embedding layer to extract 2D information from the images. We propose hybrid ViX – a combination of linear attention and convolutional layers for image classification – and compare its performance

¹Codes available at <https://github.com/pranavphoenix/VisionXformer>

with standard ViT. We replace the linear embedding layer in ViX by three convolutional layers, creating the same number of feature maps as the embedding dimension. The size of each feature map is the same as that of the input image. The hybrid ViX architecture is shown in Figure 3. The convolutional layers were also added to FNet and MLP mixer models to observe their performance.

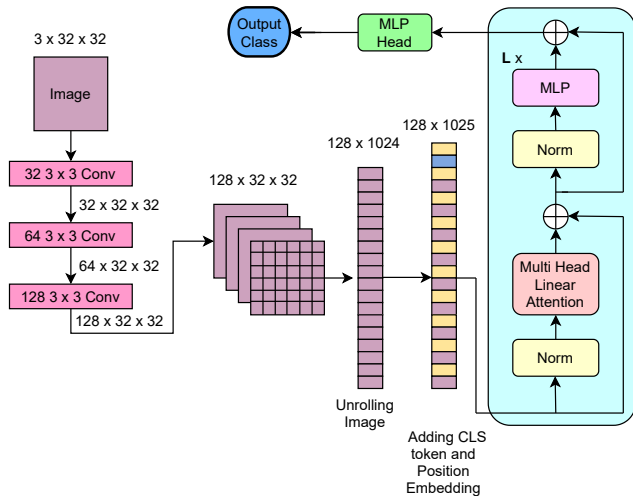


Figure 3. Hybrid ViX architecture

We also compare the performance of other transformer architectures such as LeViT, CvT, PiT, and CCT by replacing quadratic attention with linear ones, creating LeViX, CCX, CvX, and PiX. In our notation, we replace the X in X-former models with the name of the linear attention used, for example, CCT model that uses Performer becomes CCP (Compact Convolutional Performer). Addition of the prefix *hybrid* means that the convolutional layer has replaced the linear embedding layer, i.e, hybrid PiX is created by replacing the linear embedding layer with convolutional ones. LeViT adds relative positional bias when it computes the full attention matrix. Since, linear attention mechanism does not compute this matrix, we could not add the relative positional bias in LeViX. Therefore, we removed the positional bias in LeViT so that the comparison between LeViT and LeViX is justified. Subsequently, we compare the performance of ViX and hybrid ViX models by replacing 1D learnable position embedding with a rotary positional embedding.

4. Experiments

The CIFAR-10 [14] and the Tiny ImageNet [15] datasets were used for our experiments. Since the image size in the former is 32x32, we have a sequence length of 1024 upon unrolling the images and an additional class token was added for classification, making the final sequence length 1025. All models used in our experiments have 128-

dimensional embedding, 4 layers, 4 heads, 256 as MLP dimension, and 0 dropout to ensure uniformity in comparison in line with previous studies.² We used 64 benchmark points in Nyströmformer, a local window size of 256 with ReLU nonlinearity as the kernel function in Performer, and used the same projection for keys and values in Linformer. The Adam optimizer with $\alpha = 0.001$ (learning rate), $\beta_1 = 0.9$ and $\beta_2 = 0.999$ were used for computing running averages of gradient and its square, $\epsilon = 10^{-8}$, and 0.01 as weight decay coefficient were used. We used automatic mixed precision in PyTorch during training to make it faster and consume less memory. Experiments were done with 16 GB Tesla P100-PCIe and Tesla T4 GPUs available in Kaggle and Google Colab. Number of model parameters, model memory footprint (size) and GPU usage for a batch size of 64 are reported along with top-1 % and top-5% accuracy from best of three runs.

In the hybrid ViT and ViX models, we used three convolutional layers with 32, 64, and 128 3x3 kernels respectively in each stage with stride of one with padding to generate 128 feature maps of size 32x32. We used four-head, four-layer ViT architecture with patch dimension of 128, and MLP dimension of 256 for this experiment. For the comparison between LeViT and LeViX, we used a single stage model with three convolutional layers for generating the embedding. For CCT and CCX, we used learnable position embedding and 3x3 convolutional and pooling kernels having stride equal to 1 and padding. For CvT and CvX, we used one attention module in the first stage, one attention module in the second stage and 2 attention modules in the third stage with 3x3 convolutions in each stage. For the PiT, PiX, and hybrid PiX, we used a 4-layer transformer, down-sampling the sequence after every two layers.

We also used the Tiny ImageNet dataset to validate the results obtained from CIFAR-10. We set the patch size of ViX models and stride for the initial convolutional layer in hybrid ViX models as two so that the 64x64 images in the Tiny ImageNet dataset gets unrolled to a sequence of length 1024. The final classification layer was replaced to handle classification of 200 classes instead of 10 classes. All other design parameters were similar to those used for the CIFAR-10 dataset.

5. Results

5.1. ViX compared to ViT

Compared to the vanilla transformer (ViT), the Vision Nyströmformer (ViN) performs $\approx 16\%$ better, Vision Performer (ViP) performs $\approx 3\%$ better, and Vision Linformer (ViL) performs $\approx 2\%$ better for image classification of CIFAR-10 dataset, as show in Table 1. Among the ViX

²The codes for the implementation of all models were adapted from <https://github.com/lucidrains/vit-pytorch>

architectures, ViP and ViN use almost the same number of parameters as the quadratic attention, but ViL uses a significantly lower number of parameters. ViT also requires the largest share of GPU memory and storage space among all models tested. All X-former models (blue) consume less than half the storage space used by ViT (red) and use less than one-third of ViT’s GPU memory for achieving comparable performance as shown in Figure 4. Similar improvements in performance at reduced GPU RAM consumption were also observed on classification of Tiny ImageNet dataset, as shown in Table 2.

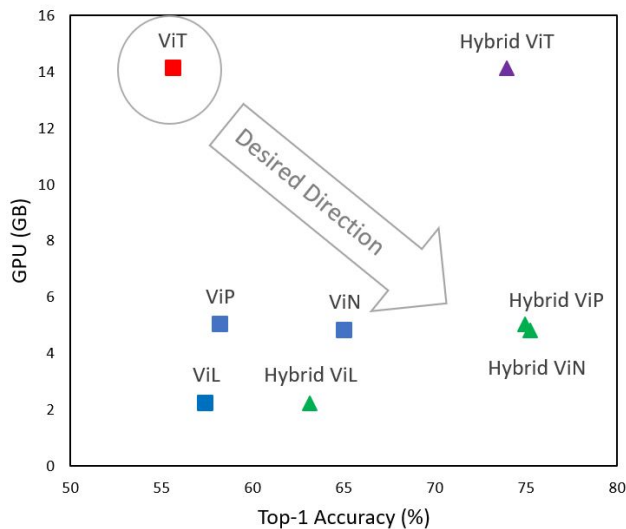


Figure 4. Comparison of classification accuracy and GPU usage by various models on CIFAR-10 for a batch size of 64

FNet has the lowest number of parameters since it completely replaces the attention module with the fast Fourier transform that does require trainable parameters. The four-layered FNet performs $\approx 10\%$ worse than ViT but uses only half the number of parameters, one-third of its memory, and only one-tenth of the GPU RAM for image classification on the CIFAR-10 dataset, as shown in Table 1. The 4 layered MLP mixer performs $\approx 7\%$ better, but uses 16 times the number of parameters that a ViT uses. However, an MLP mixer uses less than half the memory and one-tenth of the GPU RAM used by a ViT, making it a more efficient alternative to transformers. FNet performs ≈ 4 percentage points better than MLP mixer in classification of Tiny ImageNet dataset using one-tenth the GPU RAM used by ViT as shown in Table 2. The fractional GPU RAM usage by these models will enable us to use deeper models compared to ViT for a fixed GPU budget.

5.2. Hybrid ViX compared to ViX

As we can see from Table 1, replacing the initial linear embedding layer with convolutional layers increases the number of parameters slightly for all base architectures. We

also observe a slight increase in the model size and a negligible increase in the GPU usage. However, these small increases in parameters and computational resources is compensated by an appreciable improvement in performance of $\approx 20\%$ across all models for classification of CIFAR-10 dataset. Performer (Hybrid ViP) and transformer (Hybrid ViT) models both gained $\approx 28\%$ accuracy, while the Nyströmformer (Hybrid ViN) gained $\approx 16\%$. The Hybrid ViP and the Hybrid ViN outperformed the Hybrid ViT by about 1 and 2 percentage points, respectively, using only one-third of the GPU RAM and half the memory of a ViT. Even though the Hybrid Vision Linformer (ViL) underperformed compared to the other mechanisms, its computational cost is one-sixth of ViT and it also has the lowest number of parameters. The Hybrid ViN performed better than the other models in terms of the top-1 accuracy.

Adding convolutional layers to FNet improves its performance by $\approx 28\%$ for CIFAR-10 dataset while it improves by $\approx 18\%$ for Tiny ImageNet dataset. It has the smallest number of parameters and consumes less memory than other ViX models, making it an attractive alternative compared to transformer models. The higher performance of FNet compared to the Hybrid ViL despite having lesser number of parameters suggest that the weaker performance of Hybrid ViL is not due to low parameters, but the approximating mechanism used to compute attention linearly which is not suitable for vision tasks. The classification accuracy of the MLP mixer increased only marginally with the addition of convolutional layers for CIFAR-10 dataset, but increased by 4 percentage points for Tiny ImageNet dataset.

Figure 4 shows clearly that all ViX (blue squares) and hybrid (triangles) models outperform the ViT (red). The linear attention of ViX models (blue) using Performer (ViP) and Nyströmformer (ViN) outperform the quadratic attention of ViT. Further, the graph shows that all hybrid models (green and purple), which use convolutional layers for generating the embedding, outperform the models that use a linear layer (red and blue) by a large margin. Even when the convolutional layers replace the linear embedding layers, the ViX models (green) outperform normal attention in ViT (purple). This significant increase in accuracy may be attributed to the convolutional layers providing low-level features such as edges, corners, and bars to the attention layers in the low data regime. The availability of these inductive priors from convolutional layers helps the transformer learn to classify images using less data.

Adding convolutional layers significantly improved the performance of models in classification of Tiny ImageNet dataset. We can see from Table 2 that hybrid ViP and hybrid ViN achieve the accuracy of hybrid ViT with less than one-third the GPU RAM. These results holds potential for using hybrid X-formers for processing high resolution images at smaller patch sizes without exhausting GPU memory.

Model	Parameters	Size (MB)	Top-1 Acc. (%)	Top-5 Acc. (%)	GPU (GB)
Vision Transformer (ViT)	530,442	209.39	56.81	94.82	14.1
Vision Performer (ViP)	531,978	85.15	58.23	94.89	5.0
Vision Linformer (ViL)	415,754	77.70	57.45	95.11	2.2
Vision Nyströmformer (ViN)	530,970	86.62	65.06	96.42	4.8
FNet	267,786	60.12	50.87	93.22	1.5
MLP Mixer	8,533,002	90.59	60.33	95.79	1.4
Hybrid Vision Transformer (ViT)	623,178	211.47	73.94	98.08	14.1
Hybrid Vision Performer (ViP)	624,714	87.23	74.98	98.41	5.0
Hybrid Vision Linformer (ViL)	508,490	79.78	63.17	95.75	2.2
Hybrid Vision Nyströmformer (ViN)	623,706	88.70	75.26	98.39	4.8
Hybrid FNet	360,522	62.20	64.34	96.95	1.6
Hybrid MLP Mixer	8,625,738	92.67	61.58	96.26	1.4

Table 1. Test accuracy and various requirements of models compared on the CIFAR-10 dataset for a batch size of 64

Model	Parameters	Size (MB)	Top-1 Acc. (%)	Top-5 Acc. (%)	GPU (GB)
Vision Transformer (ViT)	556,104	209.59	26.43	51.82	14.1
Vision Performer (ViP)	557,640	85.35	28.17	53.93	4.4
Vision Linformer (ViL)	441,416	77.90	24.38	49.31	2.0
Vision Nyströmformer (ViN)	556,632	86.82	30.17	55.97	3.6
FNet	293,448	60.32	22.75	49.85	1.5
MLP Mixer	8,558,664	90.79	18.47	42.16	1.2
Hybrid Vision Transformer (ViT)	647,688	211.60	34.48	60.79	14.1
Hybrid Vision Performer (ViP)	649,224	87.36	34.13	60.42	4.4
Hybrid Vision Linformer (ViL)	533,000	79.91	26.90	52.88	2.2
Hybrid Vision Nyströmformer (ViN)	648,216	88.83	34.05	60.80	3.6
Hybrid FNet	385,032	62.33	26.39	51.86	1.6
Hybrid MLP Mixer	8,650,248	92.80	22.76	47.33	1.2

Table 2. Test accuracy and various requirements of models compared on the Tiny ImageNet dataset for a batch size of 64

5.3. ViX compared to other Vision Transformers

We can see from Table 3 that LeViX models using Performer (LeViP) and Nyströmformer (LeViN) outperform LeViT by about 1% and 3% respectively for classification of CIFAR-10 dataset. Even though all the models have similar number of parameters, the LeViT model takes more than twice the memory and thrice the GPU RAM compared to LeViP and LeViN models. Similar performance gain is observed in the case of CCT and CvT, where Performer (CCP and CvP) and Nyströmformer (CCN and CvN) perform better than CCT and CvT using less than half its memory and one-third of GPU RAM. The same trend is observed in Table 4 for classification of Tiny ImageNet dataset where LeViP outperforms LeViT by $\approx 8\%$, CCN and CCP outperform CCT by $\approx 3\%$, and CvN and CvP outperform CvT by $\approx 12\%$. This shows that replacing quadratic attention with Performer and Nyströmformer attention will improve the performance in long sequence image classification using lesser memory and GPU RAM. These results indicate the possibility of training transformer models in vision with

limited resources and less data using X-formers.

We also observe from Table 5 that replacing the quadratic attention in pooling-based image transformer (PiT) by ViX improves its performance. Nyströmformer (PiN) and Performer (PiP) improves the accuracy by about 9% and 3% respectively using one-third the GPU RAM and half the memory compared to PiT. We also observe the similar trend from previous sections where a significant increase in accuracy was achieved by replacing the linear embedding layer with convolutional ones. An average increase of 10% in classification accuracy was observed in Hybrid PiX models compared to PiX models.

5.4. ViX with Rotary Position Embedding

Table 6 shows that replacing the 1D learnable position embedding with Rotary Position Embedding (RoPE) further increases the performance of ViX and hybrid ViX models. We observed that using RoPE did not change the number of parameters in these models, but slightly increased the size and GPU usage. The trend of low GPU usage and memory

Model	Parameters	Size (MB)	Top-1 Accuracy (%)	Top-5 Accuracy (%)	GPU (GB)
LeViT	624,970	213.15	77.06	98.46	13.9
LeViP	662,410	77.14	79.50	98.91	4.9
LeViL	506,186	69.69	63.17	96.13	2.3
LeViN	621,402	77.13	77.81	98.61	4.3
CCT	905,547	225.47	82.23	99.04	14.7
CCP	907,083	101.48	82.48	99.06	4.3
CCL	790,859	94.04	80.05	98.92	3.5
CCN	906,075	101.48	83.36	99.07	4.2
CvT	1,099,786	173.21	79.93	99.02	12.8
CvP	827,914	100.17	83.19	99.20	4.8
CvL	711,690	92.73	72.58	97.81	2.6
CvN	826,906	100.17	83.26	99.14	4.3

Table 3. Comparison of different convolutional transformers without and with linear attention mechanisms on CIFAR-10 dataset

Model	Parameters	Size (MB)	Top-1 Accuracy (%)	Top-5 Accuracy (%)	GPU (GB)
LeViT	649,480	213.28	38.29	65.23	14
LeViP	646,920	77.27	41.49	68.39	4.9
LeViL	530,696	69.89	26.52	51.85	3.3
LeViN	645,912	77.26	38.28	64.94	4.3
CCT	930,057	225.61	39.05	65.74	14.2
CCP	931,593	101.61	40.11	66.93	4.1
CCL	815,369	94.17	37.24	62.97	2.3
CCN	930,585	101.61	40.80	67.81	3.9
CvT	1,124,296	173.34	40.69	68.19	12.3
CvP	852,424	100.3	45.26	71.21	4.4
CvL	736,200	92.86	33.95	61.29	2.3
CvN	851,416	100.3	45.47	71.35	4.0

Table 4. Comparison of different convolutional transformers without and with linear attention mechanisms on Tiny ImageNet dataset

consumption by ViX and hybrid ViX compared to ViT was observed again with RoPE.

6. Conclusions and Discussion

We showed that using linear self-attention mechanisms such as Performer and Nyströmformer in place of a vanilla attention mechanism with quadratic complexity in vision transformers can overcome a significant computational bottleneck that limits the application of transformers for long sequences formed by image pixels. More importantly, linear attention mechanisms in ViX can achieve image classification accuracy comparable to ViT while using far fewer computational resources, GPU RAM, and storage memory. This result was expected as linear attention mechanisms such as Nyströmformer and Performer were able to reach the performance of vanilla transformer using half the GPU in LRA benchmark tasks which included image classification [22, 29]. Our experiments confirm that replacing quadratic attention with these linear attention mechanisms in ViT architectures also leads to significant reduc-

tion in GPU RAM and computational costs without deteriorating their performance in image classification. This will enable the usage of transformer architectures for vision applications by practitioners who have resource constraints in terms of size of the GPUs (i.e., RAM and cores).

Additionally, we showed that replacing the initial fully connected embedding layer with the convolutional ones can independently improve the performance of vision transformers by providing an inductive prior that allows them to achieve higher accuracy with less training data. This result was obtained in parallel to ours by another group that showed similar trends by adding convolutional layers to vision transformer architectures, such as LeViT, CCT and CvT [5, 27, 6, 8]. We show that using convolutional layers to generate the embedding is enough to gain significant improvement in performance.

Furthermore, we showed that replacing the standard 1D learnable position embedding with rotary position embedding also showed independent improvement in classification accuracy. This trend was observed across different transformer architectures such as LeViT, CCT, CVT, and

Model	Parameters	Size (MB)	Top-1 Accuracy (%)	Top-5 Accuracy (%)	GPU (GB)
PiT	897,034	129.74	63.49	96.95	14.7
PiN	897,562	65.81	69.41	97.68	5.7
PiP	898,570	64.09	65.83	97.28	5.2
Hybrid PiT	989,770	131.80	73.34	98.11	14.8
Hybrid PiN	990,298	67.87	74.00	98.03	5.7
Hybrid PiP	991,306	66.15	71.37	97.88	5.2

Table 5. Comparison of performance by replacing attention with X-formers in PiT for classification of CIFAR-10 dataset

Model	Top-1 Accuracy (%)	Top-5 Accuracy (%)	GPU (GB)
ViT	69.32	97.68	14.4
ViP	60.74	95.62	5.2
ViL	55.88	94.61	2.7
ViN	68.36	97.28	5.2
Hybrid ViT	76.90	98.60	14.5
Hybrid ViP	76.86	98.37	5.2
Hybrid ViL	64.40	96.63	2.7
Hybrid ViN	74.38	97.64	5.2

Table 6. Performance of ViX and Hybrid ViX models using Rotary Position Embedding on CIFAR-10 dataset

PiT when we implemented these changes in them. Alternative architectures that are efficient in token mixing, such as FNet and MLP mixer, also hold promise in vision applications as they require only one-tenth of the GPU RAM used by a vanilla ViT. Using ViX, FNet or MLP mixer will enable the use of deeper models with a fixed GPU RAM budget compared to vanilla ViT. Addition of convolutional layers for generating the embedding significantly improved the performance of FNet, but such an increase in accuracy was not observed with MLP mixer. This shows that making a hybrid MLP mixer does not justify the increase in space and parameter count. So, we do not recommend using convolutional layers in the MLP mixer.

Undoubtedly, further experiments with hyper-parameter tuning, using different positional embedding, alternative mixing models, and image datasets and tasks can lead to more insights for improving the efficiency and accuracy in low data, low GPU RAM regime. However, our results point to several significant directions for developing alternatives to purely convolutional architectures for vision. While the use of transformer architectures for vision has produced tantalizing results in terms of accuracy, these have come at orders of magnitude higher costs in terms of training data, floating-point operations, GPU RAM, hardware costs, and power consumption [17]. This has put training transformer-type architectures out of the reach of all researchers, except those in a select few organizations with massive budgets, while their monetization is still in question. On the other hand, in spite of the theoretical guarantee of a fully connected neural network with a single hidden layer being able to model any function [13], proposals of neural architec-

tures that exploit domain-specific inductive biases have resulted in usable increases in accuracy as well as decreases in computational and data requirements. Our work suggests that this quest is far from over, and architectural innovation can democratize the ability to train neural architectures from scratch with state-of-the-art performance.

References

- [1] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019.
- [2] Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers, 2021.
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
- [4] 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, 2021.
- [5] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet’s clothing for faster inference, 2021.
- [6] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers, 2021.

- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [8] Byeongho Heo, Sangdoon Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers, 2021.
- [9] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.
- [10] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5mb model size, 2016.
- [11] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention, 2020.
- [12] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey, 2021.
- [13] Andrei Nikolaevich Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. In *Doklady Akademii Nauk*, volume 114, pages 953–956. Russian Academy of Sciences, 1957.
- [14] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.
- [15] Ya Le and X. Yang. Tiny imagenet visual recognition challenge. 2015.
- [16] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms, 2021.
- [17] Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In *International Conference on Machine Learning*, pages 5958–5968. PMLR, 2020.
- [18] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.
- [19] Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Do large scale molecular language representations capture important structural information?, 2021.
- [20] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2021.
- [21] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.
- [22] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers, 2020.
- [23] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey, 2020.
- [24] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision, 2021.
- [25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
- [26] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020.
- [27] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers, 2021.
- [28] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better, 2021.
- [29] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention, 2021.