ReduceFormer: Attention with Tensor Reduction by Summation

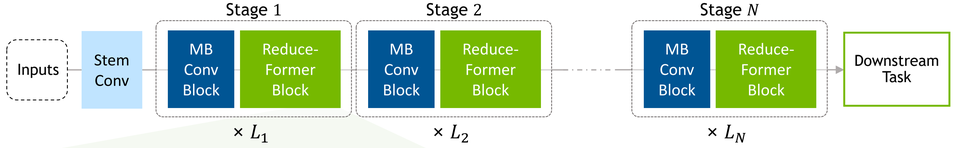
John Yang, Le An, Su Inn Park

NVIDIA

In this paper, we propose **ReduceFormer**, a family of vision models exclusively harnessing only basic operations such as element-wise multiplication and global summation to model both local and global feature relationships. The advantages of the proposed approach over previous methods are mainly twofold:

- Reduced Model Complexity: we eliminate the use of matrix multiplication and expensive operations such as Softmax as in typical attention blocks, leading to a much simpler model structure.
- Efficiency in Inference: The operations in the proposed series of models can utilize well-optimized implementations on modern deep learning accelerators such as GPU, resulting in improved efficiency in latency, throughput, and memory footprint.

ReduceFormer Framework

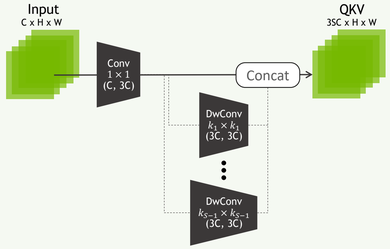


ReduceFormer Block

Our framework uses the ReduceFormer block for enhanced global information learning, incorporating efficient feature-reduction techniques like global summation and element-wise operations. ReLU-based attention processes, followed by an FC layer, handle initial multi-scale local context extraction and subsequent non-local feature relations.

Multi-Scale Local Feature Encoding

Our method employs depth-wise convolution operators with various kernel sizes to address the limitation of ReLU-based attention in capturing local context, without significantly increasing the model's parameter footprint.



ReduceFormer Attention

ReLU linear attention[1] allows latency improvement against Softmax attention to an extent, but still suffers from costly computational complexity caused by matrix multiplications:

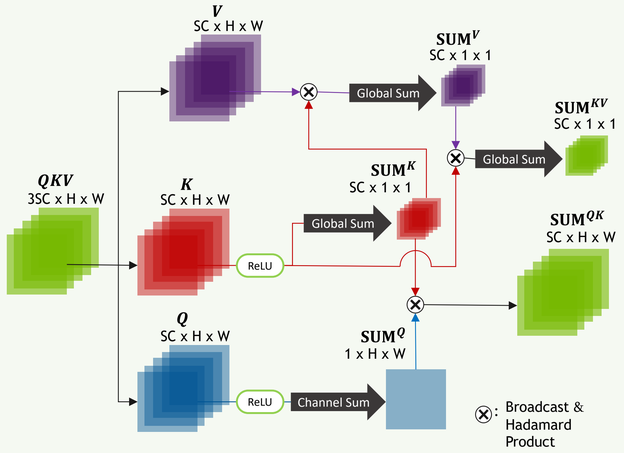
$$O_i = \frac{\mathrm{ReLU}(Q_i)\sum_{j=1}^{N}\mathrm{ReLU}(K_j)^T V_j}{\mathrm{ReLU}(Q_i)\sum_{j=1}^{N}\mathrm{ReLU}(K_j)^T}$$

for i-th feature in the output feature O.

To address this, our method employs repeated global summations and element-wise multiplications to approximate and bypass the inner product calculation of K and V, improving computational efficiency while emphasizing global feature projection. For activation, each pixel is divided by the sum of all features, summarized in the computation of the ReduceFormer attention block:

$$O_i = \frac{\mathrm{ReLU}(Q_i)\mathrm{SUM}^{KV}}{\mathrm{SUM}_i^{QK}}$$

where each term in the right-hand is computed as shown in the Figure on the right.



Experiments

- ImageNet Dataset
- Platforms
  - NVIDIA DRIVE Orin[2]
  - L40 GPU
- TensorRT[3] with FP16
- Metrics
  - Model Sizes
  - Model Complexity
  - Accuracy
  - Orin
    - Latency
    - Memory BW
  - L40
    - Throughput

Models	#Params	MACs	Top1 Acc ↑ (%)	NVIDIA DRIVE Orin SoC FP16 Latency ↓ (ms)	NVIDIA DRIVE Orin SoC Avg Mem BW ↓ (MB/image)	L40 GPU Throughput ↑ (images/sec)
EfficientViT-B1 (r224) [1]	9.1M	0.53G	79.4	0.90	27.48	3067
EfficientViT-B1 (r256) [1]	9.1M	0.69G	79.9	0.98	32.96	2976
EfficientViT-B1 (r288) [1]	9.1M	0.87G	80.4	1.14	36.84	2817
ReduceFormer-B1 (r224)	9.0M	0.52G	79.3	**0.68 (32%↓)**	**26.02 (6%↓)**	**4149 (35%↑)**
ReduceFormer-B1 (r256)	9.0M	0.67G	80.1	**0.73 (34%↓)**	**30.50 (8%↓)**	**4049 (36%↑)**
ReduceFormer-B1 (r288)	9.0M	0.85G	80.6	**0.87 (31%↓)**	**36.30 (2%↓)**	**3731 (32%↑)**
CoAtNet-0 [4]	25M	4.2G	81.6	2.71	155.25	1742
ConvNeXt-T [13]	29M	4.5G	82.1	2.11	98.55	2247
EfficientViT-B2 (r256) [1]	24M	2.1G	82.7	1.86	84.38	1931
EfficientViT-B2 (r288) [1]	24M	2.7G	83.1	2.23	106.87	1815
ReduceFormer-B2 (r256)	24M	2.1G	82.6	**1.41 (32%↓)**	**79.14 (7%↓)**	**2625 (36%↑)**
ReduceFormer-B2 (r288)	24M	2.8G	83.0	**1.68 (33%↓)**	**98.75 (8%↓)**	**2439 (34%↑)**
Swin-B [11]	88M	15G	83.5	4.20	319.98	1142
CoAtNet-1 [4]	42M	8.4G	83.3	4.65	258.47	980
ConvNeXt-S [13]	50M	8.7G	83.1	4.34	209.88	1274
EfficientViT-B3 (r224) [1]	49M	4.0G	83.5	2.93	152.77	1267
EfficientViT-B3 (r256) [1]	49M	5.2G	83.8	3.26	186.78	1203
ReduceFormer-B3 (r224)	48M	3.9G	83.4	**2.22 (31%↓)**	**138.65 (10%↓)**	**1742 (37%↑)**
ReduceFormer-B3 (r256)	48M	5.1G	83.6	**2.43 (33%↓)**	**173.36 (8%↓)**	**1631 (36%↑)**
CoAtNet-2 [4]	75M	16G	84.1	6.02	434.55	845
ConvNeXt-B [13]	89M	15G	83.8	5.82	351.39	1021
EfficientViT-B3 (r288) [1]	49M	6.6G	84.2	4.10	226.43	1087
ReduceFormer-B3 (r288)	48M	6.4G	84.2	**3.03 (37%↓)**	**210.89 (7%↓)**	**1464 (35%↑)**

Classification Results on ImageNet-1K data. Latency and throughput were measured with TensorRT[3] in FP16 precision on NVIDIA DRIVE Orin[2] and L40 GPU. Memory bandwidth (Mem BW) is derived from memory read and written during inference per image and averaged over 1000 runs. The percentage in parentheses is calculated with respect to its counterpart from EfficientViT[1] .

Models	Throughput (images/s) ↑ bs8	bs16	bs32
E.ViT-B1 (r224) [1]	14084	19607	23357
E.ViT-B1 (r256) [1]	12841	16949	19184
E.ViT-B1 (r288) [1]	10974	13640	14420
RF-B1 (r224)	**20202 (43%↑)**	**28120 (43%↑)**	**32128 (38%↑)**
RF-B1 (r256)	**18433 (44%↑)**	**24279 (43%↑)**	**26251 (37%↑)**
RF-B1 (r288)	**15504 (41%↑)**	**19093 (40%↑)**	**19070 (32%↑)**
E.ViT-B2 (r224) [1]	7881	9864	10873
E.ViT-B2 (r256) [1]	6866	8285	7860
E.ViT-B2 (r288) [1]	5738	6554	5851
RF-B2 (r224)	**11189 (42%↑)**	**13389 (36%↑)**	**13937 (28%↑)**
RF-B2 (r256)	**9581 (40%↑)**	**11276 (36%↑)**	**10464 (33%↑)**
RF-B2 (r288)	**7882 (37%↑)**	**8748 (34%↑)**	**7464 (28%↑)**
E.ViT-B3 (r224) [1]	4412	5313	5209
E.ViT-B3 (r256) [1]	3891	4366	3735
E.ViT-B3 (r288) [1]	3104	3319	2769
RF-B3 (r224)	**6088 (38%↑)**	**7201 (36%↑)**	**6798 (30%↑)**
RF-B3 (r256)	**5249 (35%↑)**	**5848 (34%↑)**	**5012 (34%↑)**
RF-B3 (r288)	**4177 (35%↑)**	**4392 (32%↑)**	**3717 (34%↑)**

Throughput comparison between EfficientViT[1] (E.ViT) and ReduceFormer (RF) with different batch sizes. Measured on L40 GPU with TensorRT[3] in FP16 precision.

Results indicate that ReduceFormer and EfficientViT[1] achieve similar accuracy, with ReduceFormer outperforming EfficientViT by 33% in inference latency in average on DRIVE Orin, excelling in smaller variants and maintaining a smaller memory footprint.

On the L40 GPU, ReduceFormer significantly outperforms other methods in terms of throughput for all variants at a batch size of one. Also, note that:

- average 38% performance increase compared to EfficientViT for batch sizes of 8, 16 and 32.
- a notable 44% higher throughput for the B1 variant at a batch size of 8.

This superior performance makes ReduceFormer especially advantageous for high-throughput cloud computing scenarios.

References

[1] Cai et al., "Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction", ICCV 2023.
[2] https://developer.nvidia.com/drive/agx
[3] https://developer.nvidia.com/tensorrt