THU-AM-198

Global Vision Transformer Pruning with Hessian-Aware Saliency

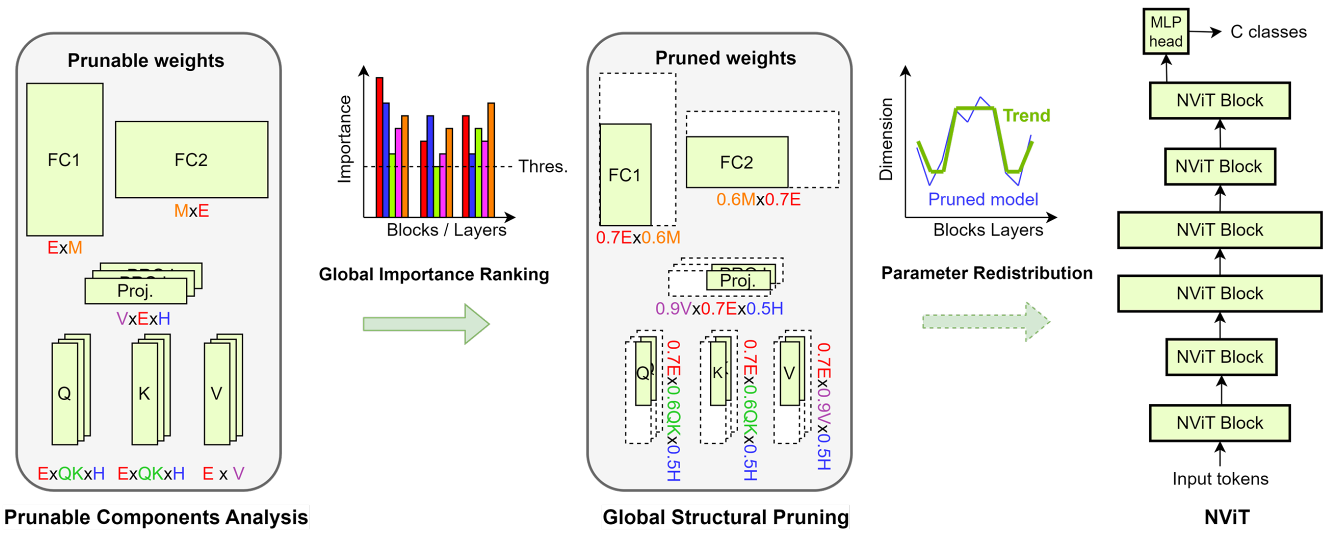
Huanrui Yang, Hongxu Yin, Maying Shen, Pavlo Molchanov, Hai Li, and Jan Kautz

NVIDIA, UC Berkeley, Duke University

NVIDIA Berkeley UNIVERSITY OF CALIFORNIA Duke

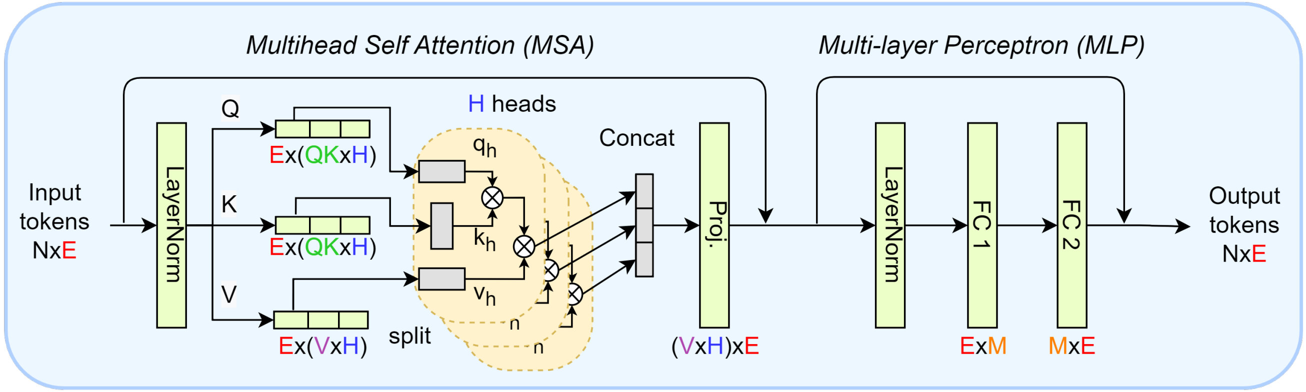
JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Overall workflow



- **1.9x** lossless DeiT-B speedup
- **+1.7%** acc @ DeiT-T latency
- **+1.4%** acc with pruning-inspired redistribution
- **6%** free speedup with Ampere

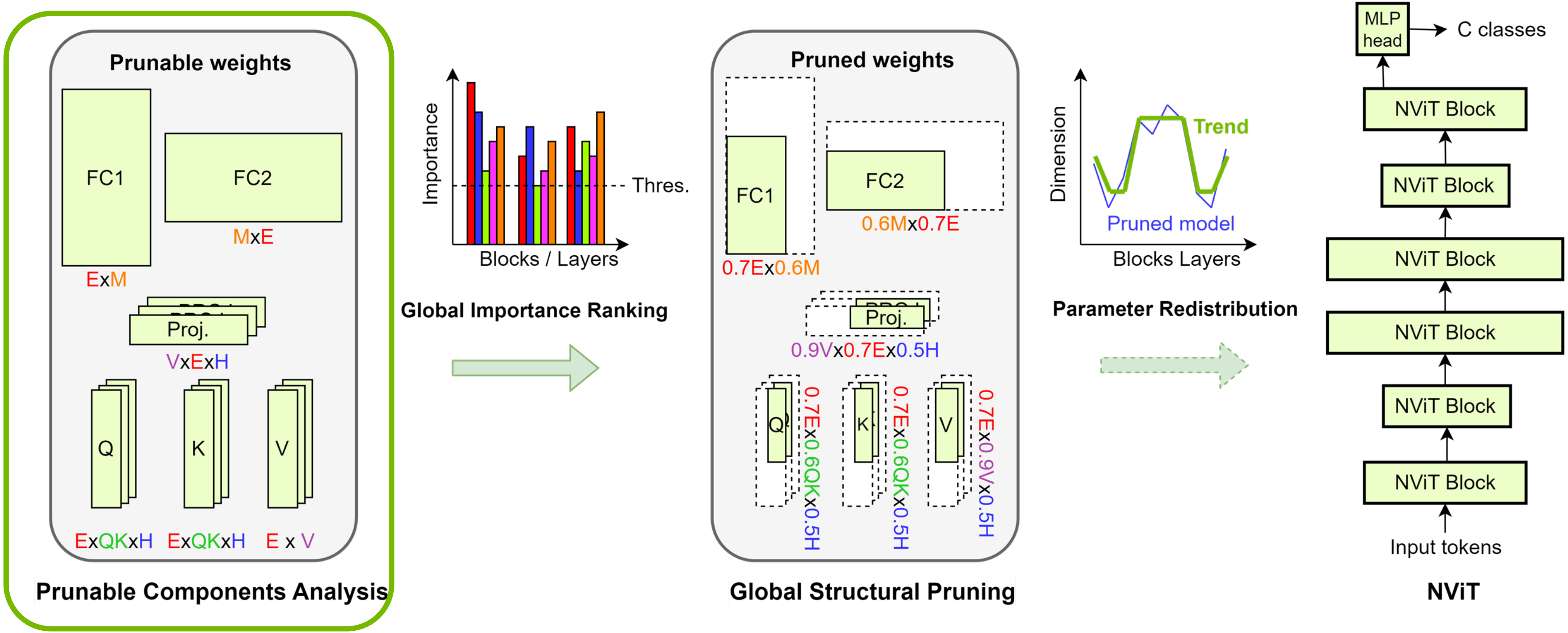
Challenges in finding efficient ViT



- Distinct architectural components with different dimensions and value ranges
- Multiple independent dimensions induce huge search space
  - Manually designed layer-wise sparsity not optimal

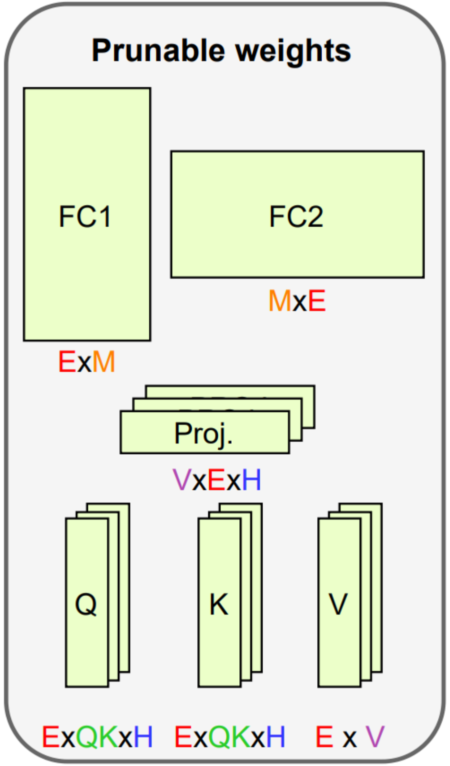
Global structural pruning required

Identifying prunable components



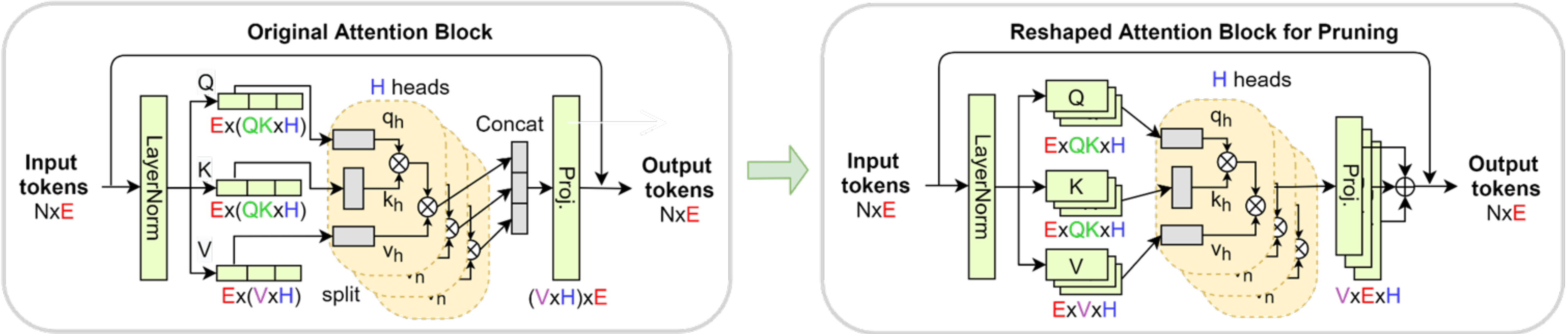
Prunable components summary

- Shared across all blocks
  - EMB: Embedding
- Indepent in each block
  - H: Number of heads
  - QK: Output dimension of Q and K projection
  - V: Output dimension of V projection
  - MLP: Hidden dimension of MLP per block



Identifying prunable components

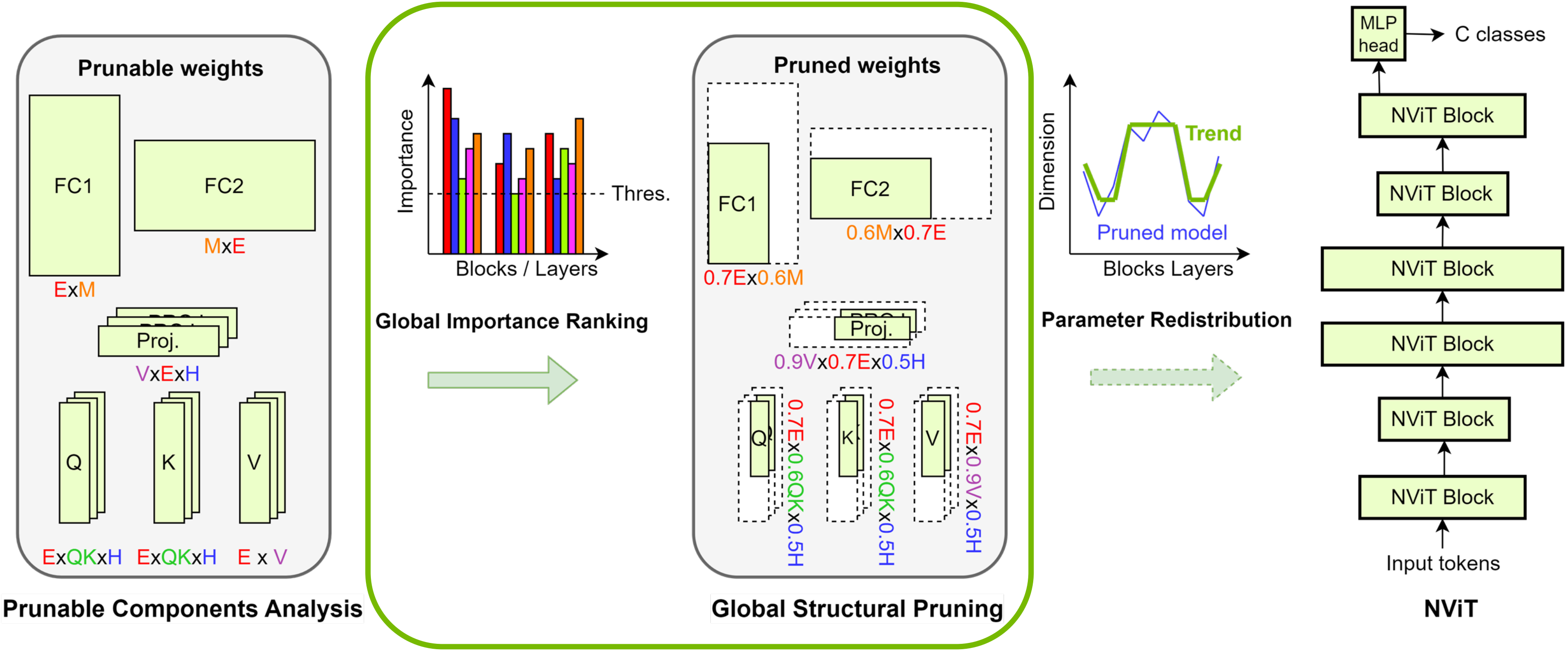
- **Insight:** Explicit head alignment
  - Imbalanced QK/V dimension in each head hurts parallelization
  - Control #head and align QK/V in each head with reshaped attention



Observation: Better utilization of latency budget

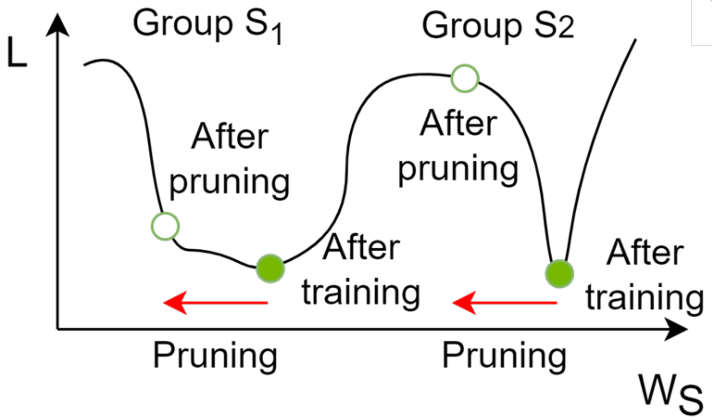
- Head alignment +0.4% accuracy than w/o alignment

Global Structural Pruning



Hessian-aware importance criteria

- Removing components with lower curvature reduces pruning loss



$$\mathcal{H}z \approx (\nabla_{gs}\mathcal{L}(gs + hz) - \nabla_{gs}\mathcal{L}(gs))/h,$$

$$\begin{aligned}\mathcal{I}_S(\mathbf{W}) &= \mathbb{E}_z \| hz \sum_{s\in \mathcal{S}} \nabla_{w_s}\mathcal{L}(w_s)\, w_s/h\|^2 \\ &= \left(\sum_{s\in \mathcal{S}}\mathcal{L}'(w_s)\, w_s\right)^2 \mathbb{E}_z z^2 \\ &= \left(\sum_{s\in \mathcal{S}}\mathcal{L}'(w_s)\, w_s\right)^2,\end{aligned}$$

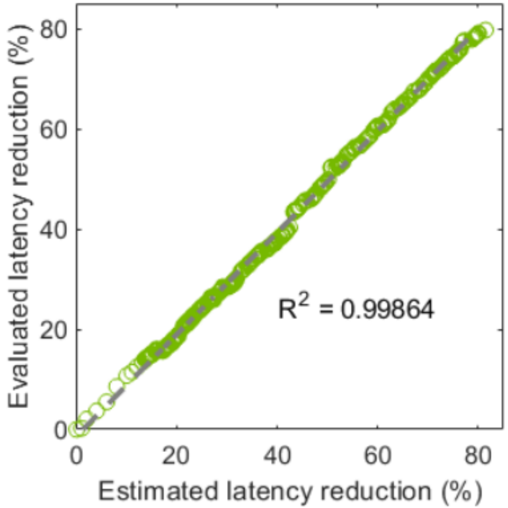
Magnitude-based criteria drops additional 40% accuracy than Hessian-aware in Base->Small compression

Latency-aware regularization

- Adjust importance score with latency reduction

$$\mathcal{I}_{\mathcal{S}}^{L}(\mathbf{W}) = \mathcal{I}_{\mathcal{S}}(\mathbf{W}) - \eta\Big(\mathrm{Lat}(\mathbf{W}) - \mathrm{Lat}(\mathbf{W}\backslash\mathcal{S})\Big)$$

- Efficient model latency estimation via latency lookup table
  - Linear interpolate between 9,000 profiled latency



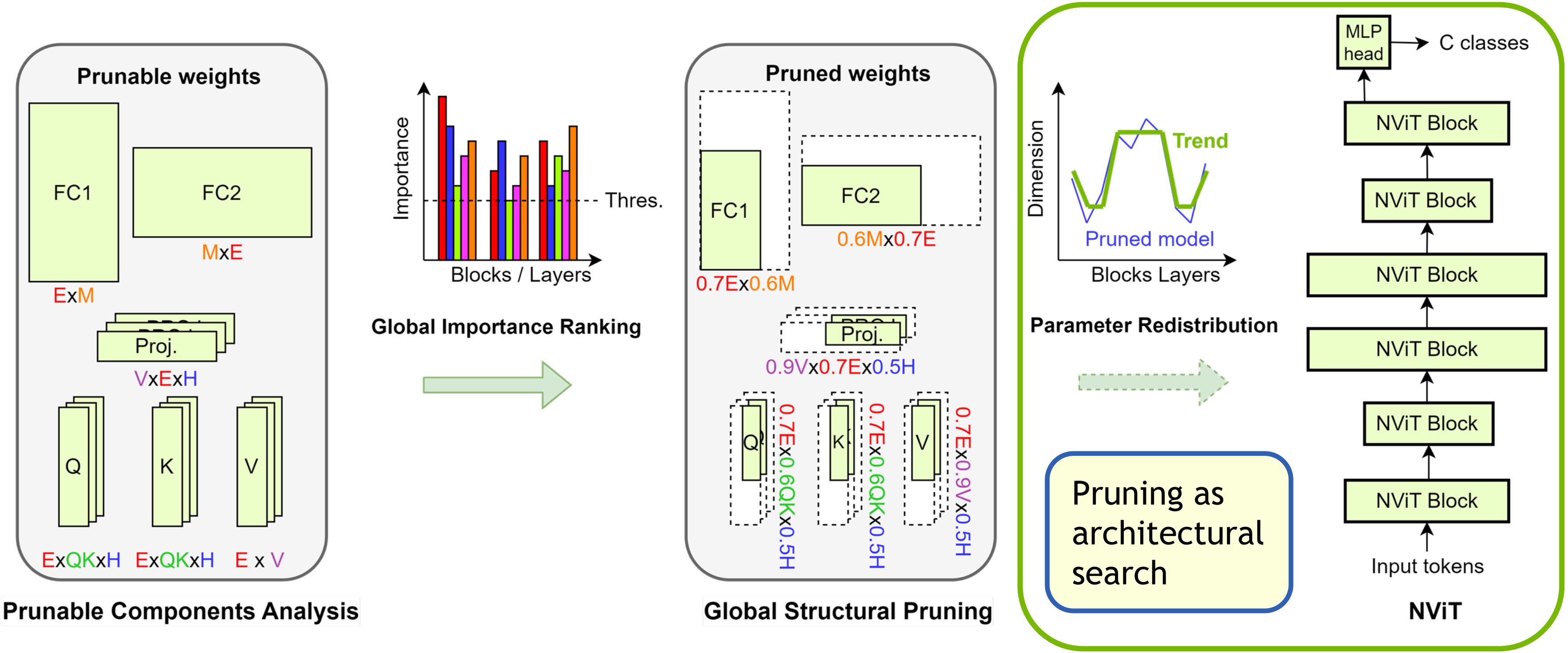
Pruning analysis on ImageNet-1K

- Lossless compression
  - 1.86x speedup over DEIT-B
- 2x speedup
  - 2x speedup with -0.4% acc
  - 1.4x faster than SWIN-S
- Base -> Small
  - +1% acc over DEIT-S
- Base -> Tiny
  - +1.7% acc over DEIT-T

Largely outperforms SOTA ViT compression methods

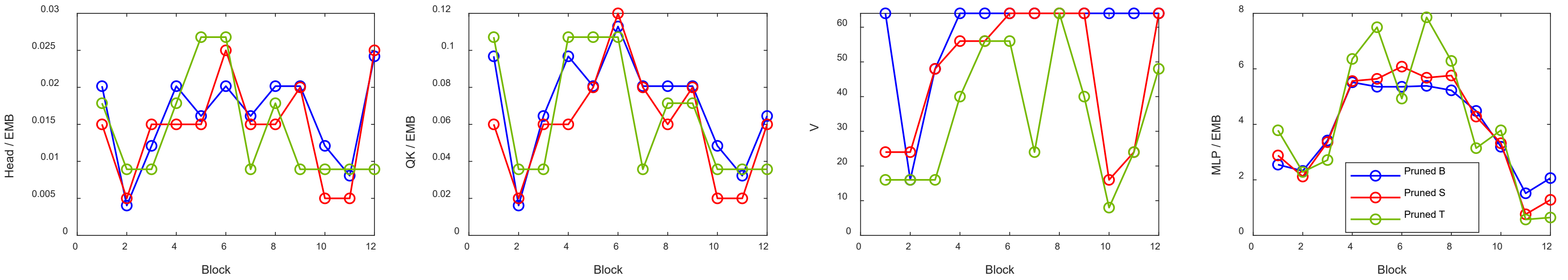
	Size (Compression)		Speedup (×)		
Model	#Para (×)	#FLOPs (×)	V100	RTX 3080	Top-1 Acc.
DEIT-B	86M (1.00)	17.6G (1.00)	1.00	1.00	83.36
SWIN-B	88M (0.99)	15.4G (1.14)	0.95	-	83.30
NViT-B	34M (2.57)	6.8G (2.57)	**1.86**	1.75	83.29
+ ASP	17M (5.14)	6.8G (2.57)	**1.86**	**1.85**	83.29
SWIN-S	50M (1.74)	8.7G (2.02)	1.49	-	83.00
NViT-H	30M (2.84)	6.2G (2.85)	**2.01**	1.89	82.95
+ ASP	15M (5.68)	6.2G (2.85)	**2.01**	**1.99**	82.95
DEIT-S	22M (3.94)	4.6G (3.82)	2.44	2.27	81.20
SWIN-T	29M (2.99)	4.5G (3.91)	2.58	-	81.30
NViT-S	21M (4.18)	4.2G (4.24)	**2.52**	2.35	**82.19**
+ ASP	10.5M (8.36)	4.2G (4.24)	**2.52**	**2.47**	**82.19**
DEIT-T	5.6M (15.28)	1.2G (14.01)	5.18	4.66	74.50
NViT-T	6.9M (12.47)	1.3G (13.55)	4.97	4.55	**76.21**
+ ASP	3.5M (24.94)	1.3G (13.55)	4.97	**4.66**	**76.21**

Exploring parameter redistribution



Trends observed in ViT pruning

- Remained dimensions under different pruning configurations

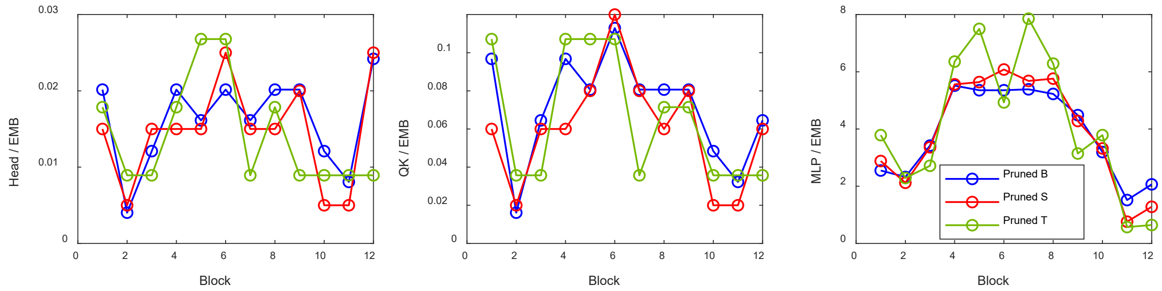


Linear scaling with EMB

- H, QK and MLP scales linearly with EMB
- V stays largely the same

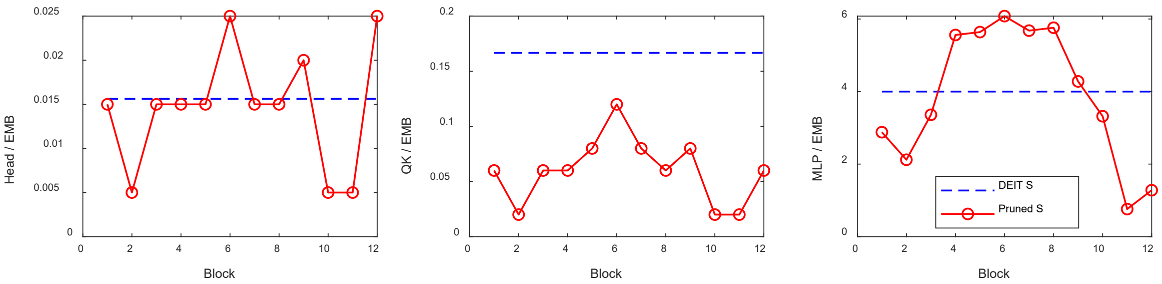
Trends observed in ViT pruning

- Block-wise parameter redistribution



- Less-more-less trend
- First and last block more important

- In-block parameter redistribution



- Less QK/V
- More MLP

Design novel architecture

Pruned models

(inspires)

Embedding-based distribution rule

Blocks	H	QK	V	MLP
DeiT	EMB/64	64	64	EMB×4
ReViT	ϵ×EMB/100	ϵ×EMB/20	64	ϵ×EMB×3

(yields)

Consistent improvements over hand-designed

Model	EMB	#Para (×)	#FLOPs (×)	Speedup	Accuracy
DeiT-S	384	22M (3.94)	4.6G (3.82)	2.29×	81.01%*
ReViT-S	384	23M (3.82)	4.7G (3.75)	2.31×	**81.22%**
DeiT-T	192	5.6M (15.28)	1.2G (14.01)	4.39×	72.84%*
ReViT-T	176	5.9M (14.64)	1.3G (13.69)	4.75×	**74.20%**

- Less-more-less trend effective for efficient ViT design
- Trade QK with MLP for higher accuracy under latency budget
- Global pruning facilitates efficient architecture discovery

Global Vision Transformer Pruning with Hessian-Aware Saliency

Thanks!
Q & A

Paper

Code

NVIDIA

Berkeley
UNIVERSITY OF CALIFORNIA

Duke

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA