QuadraNet: Improving High-Order Neural Interaction Efficiency with Hardware-Aware Quadratic Neural Networks

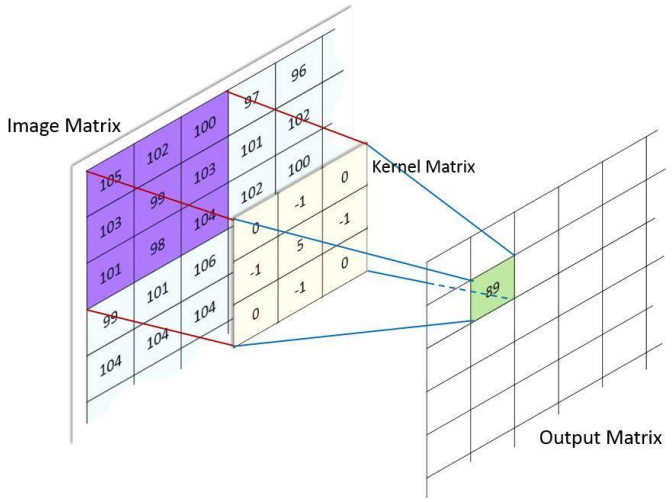
Chenhui Xu[1,2], Fuxun Yu[1], Zirui Xu[1], Chenchen Liu[3], Jinjun Xiong[2], Xiang Chen[1,4]

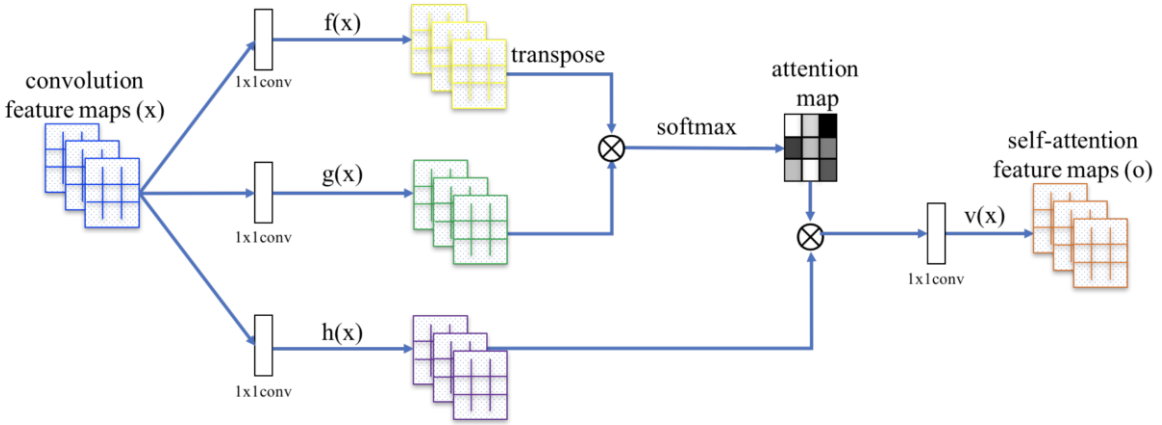
Jan 23, 2024

[1] George Mason University
[2] University at Buffalo, SUNY
[3] University of Maryland, Baltimore County
[4] Peking University

Background

- During the past decade, backbone neural network for computer vision models experienced a technological revolution.





Convolutional Neural Networks with Simple Information Extraction

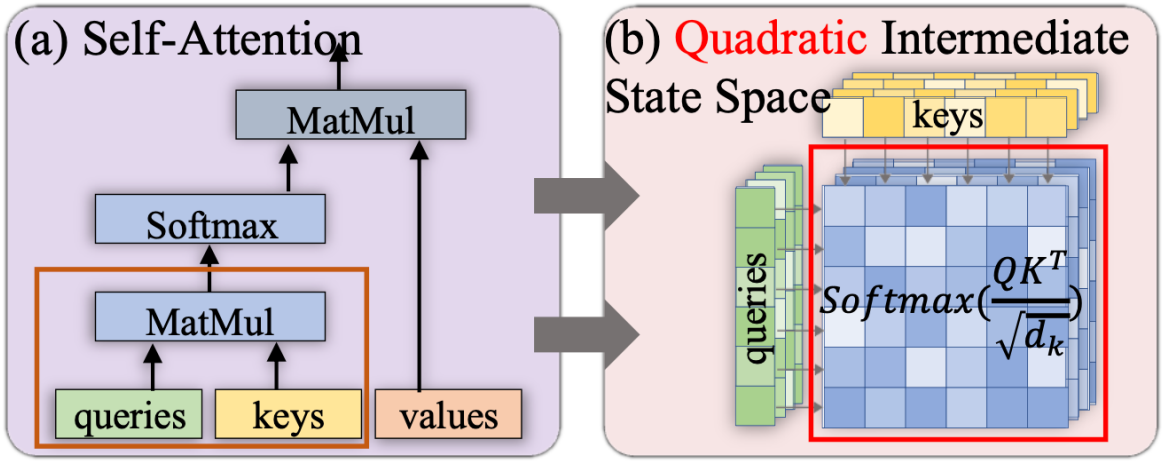
Vision Transformers with High-Order Interaction

More Resource Consuming!

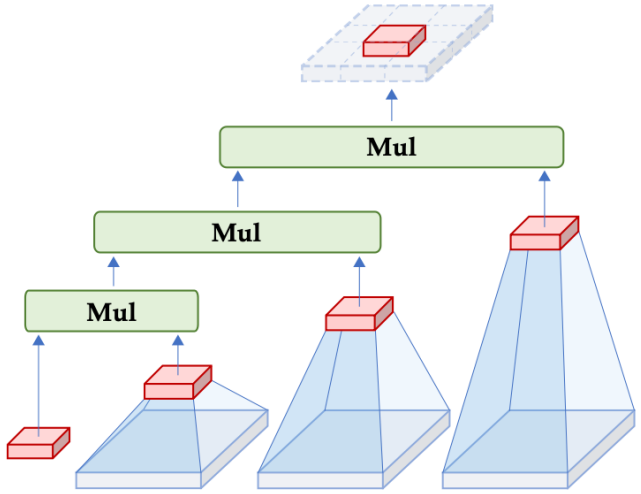
Motivation

- The High-order neural networks represented by ViT have achieved great success in algorithm performance, but their widespread deployment is limited!

$$y_{high-order}^{(i,c)} = \sum_{j \in \Omega_i} \sum_{c'=1}^{C} g(x)_{ij} T^{(c',c)} x^{(j,c')},$$



Quadratic time and space complexity



Data dependency

Motivation

- Is there a family of neural networks can fit heterologous hardware?

1. Better cognition capability	✓ High-order neural interaction
2. Less Memory Footprint	✓ Linear Complexity
3. Low latency	✓ Less data dependency
4. Scalable	✓ Block design
5. Good community support	✓ Leveraging existing accelerator and library

Method: QuadraNet Overview

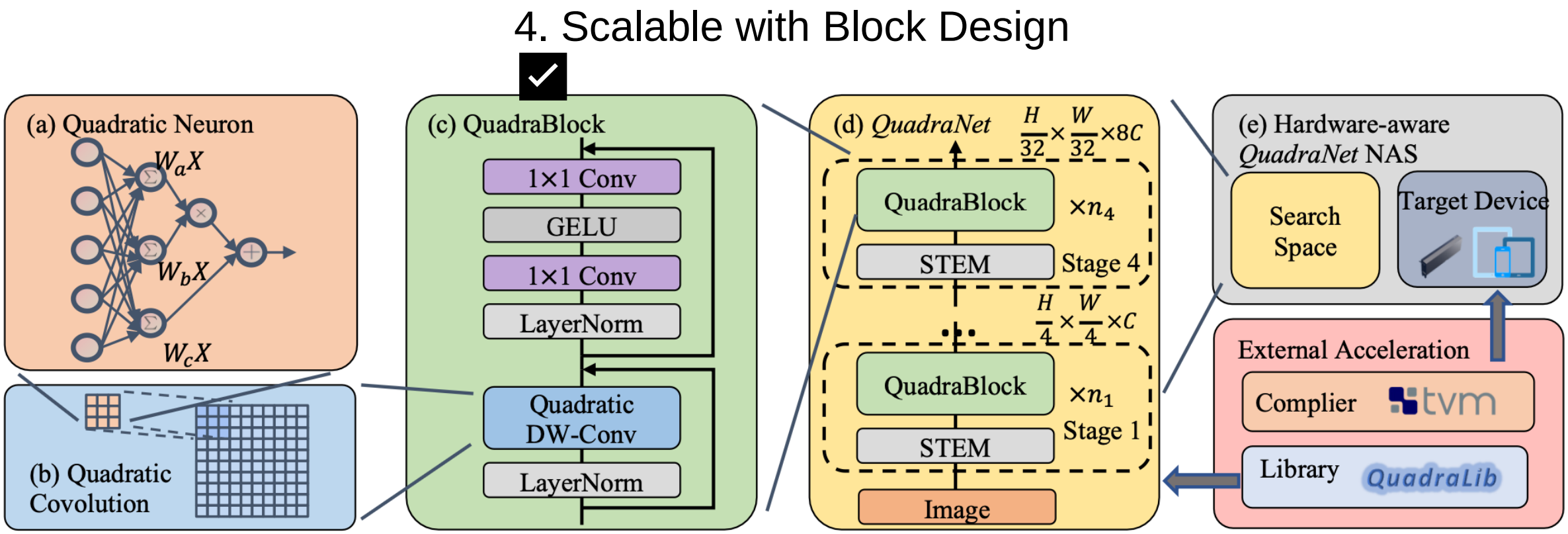


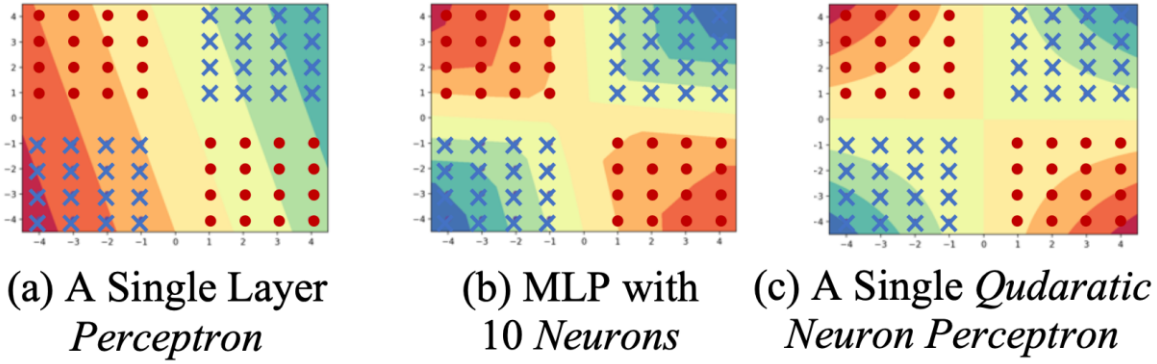
Fig. 3: A Bottom-Up Overview of *QuadraNet* (i.e., from neuron level to model level to system implementation level)

Method: Quadratic Neuron

$$y = Act(Wax + b),$$

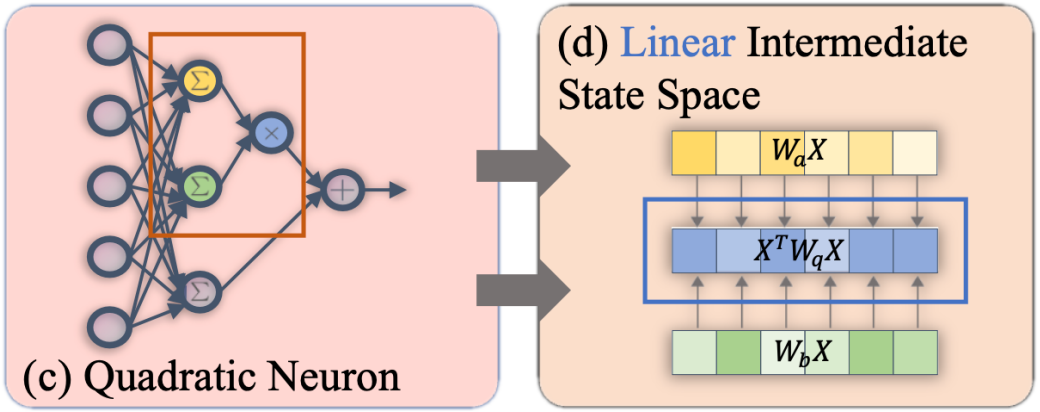
$$y = Act(x^T W_q x + W_c x + b).$$

1. ✅ Better Representation Capacity from High-order neural interaction



(a) A Single Layer *Perceptron* (b) MLP with 10 *Neurons* (c) A Single *Qudaratic Neuron Perceptron*

2. ✅ Linear Complexity



Method: Quadratic Neuron

- Quadratic neuron exhibits higher "**performance density**", especially in small model scenario.

~6% performace gain in small models!

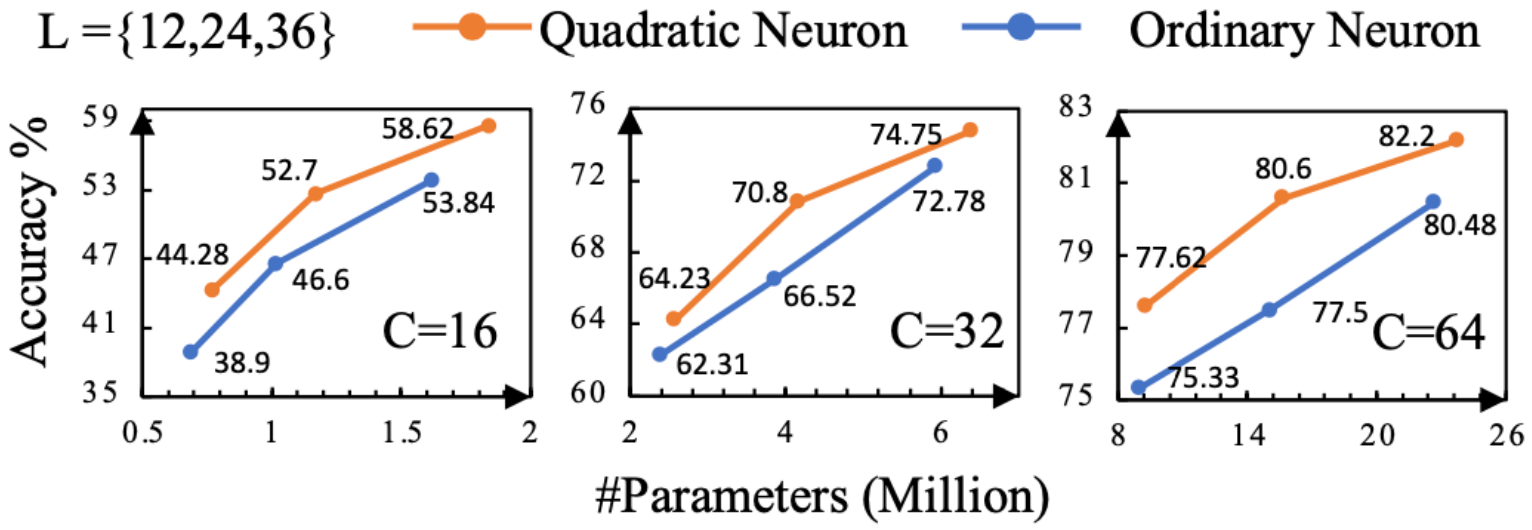
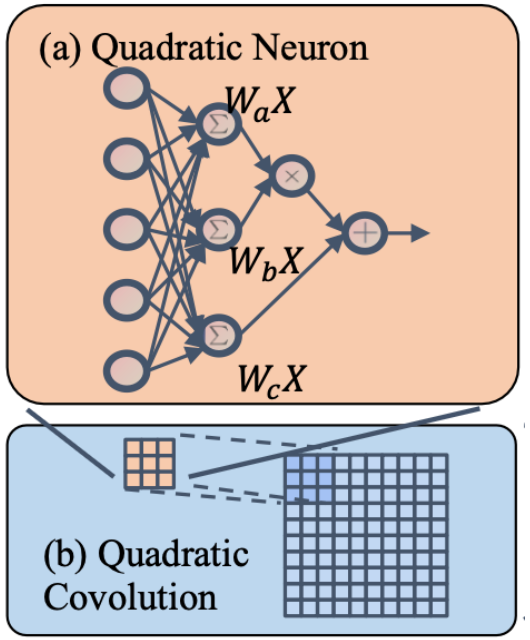


Fig. 5: Outstanding Quadratic Performance w/ Various Configurations

Method: Quadratic Neuron and Convolution

$$y = Act(x^T W_q x + W_c x + b).$$

$$y = Act(x^T W_a^T W_b x + W_c x + b).$$



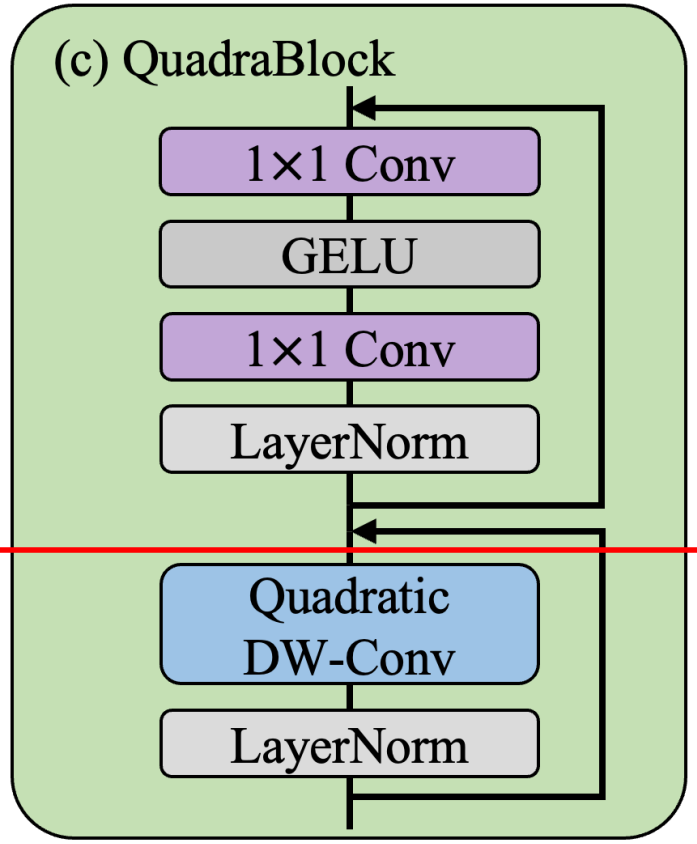
- Parameter volume and # of MACs: from n^2 to $2n$
- From Tensor perspective: $X_{out} = f_a(X_{in}) \odot f_b(X_{in}) + f_c(X_{in})$, where $f(\cdot)$ is (convolutional) operator optimized for various devices
- Fast Skip Backpropagation:

$$\frac{\partial \mathcal{L}}{\partial W_a^l} = \frac{\partial \mathcal{L}}{\partial X^{l+1}} \cdot \frac{\partial X^{l+1}}{\partial (W_a^l X^l)(W_b^l X^l)} \cdot \frac{\partial (W_a^l X^l)(W_b^l X^l)}{\partial (W_a^l X^l)} \cdot \frac{\partial (W_a^l X^l)}{\partial W_a^l}$$
$$= \frac{\partial \mathcal{L}}{\partial X^{l+1}} \cdot (W_b^l X^l) \cdot X^l,$$

5. Leveraging existing accelerator and library ✓

QuadraLib (MLSys 2022)

Method: QuadraBlock



3. Less data dependency ✔ with only one layer high-order interaction design

$$y_{Quad}^{(i,c)} = \sum_{j,k\in\Omega_i}\sum_{c'=1}^{C}(\omega_{a,i\to j}T_a^{(c,c')}x^{(j,c')})(\omega_{b,i\to k}T_b^{(c,c')}x^{(k,c')})$$

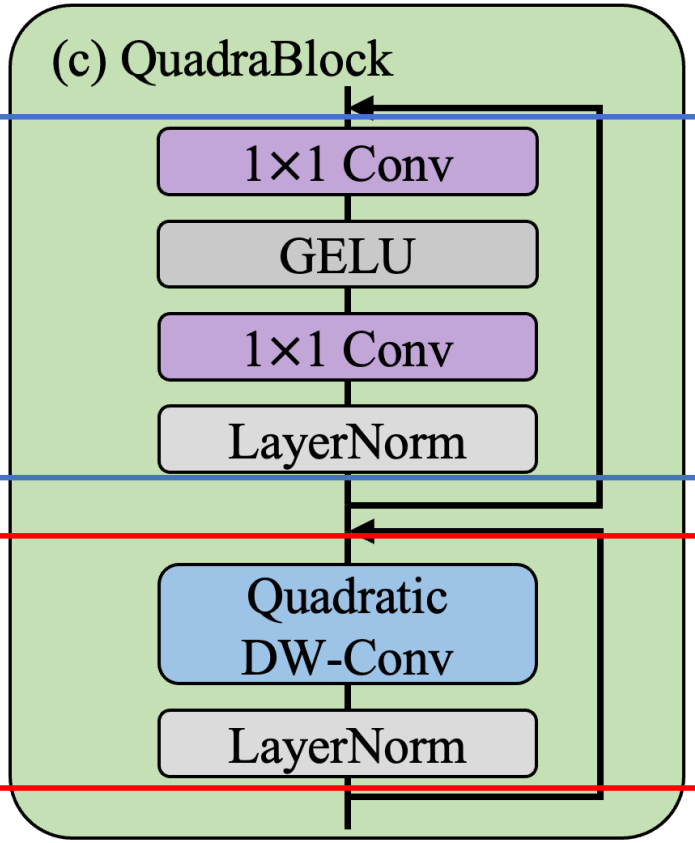
$$= \sum_{j\in\Omega_i}\sum_{c'=1}^{C}\omega_{a,i\to j}\sum_{k\in\Omega_i}\omega_{b,i\to k}x^{(k,c')}T_q^{(c,c')}x^{(j,c')}$$

$$= \sum_{j\in\Omega_i}\sum_{c'=1}^{C}q(x)_{ij}T_q^{(c,c')}x^{(j,c')},$$

where $q(x)_{ij} = \omega_{a,i\to j}\sum_{k\in\Omega_i}\omega_{b,i\to k}x^{(k,c')}$

$$y_{high-order}^{(i,c)} = \sum_{j\in\Omega_i}\sum_{c'=1}^{C}g(x)_{ij}T^{(c',c)}x^{(j,c')},$$

Method: QuadraBlock



Channel-wise information mixing
~95% of the parameters and MACs

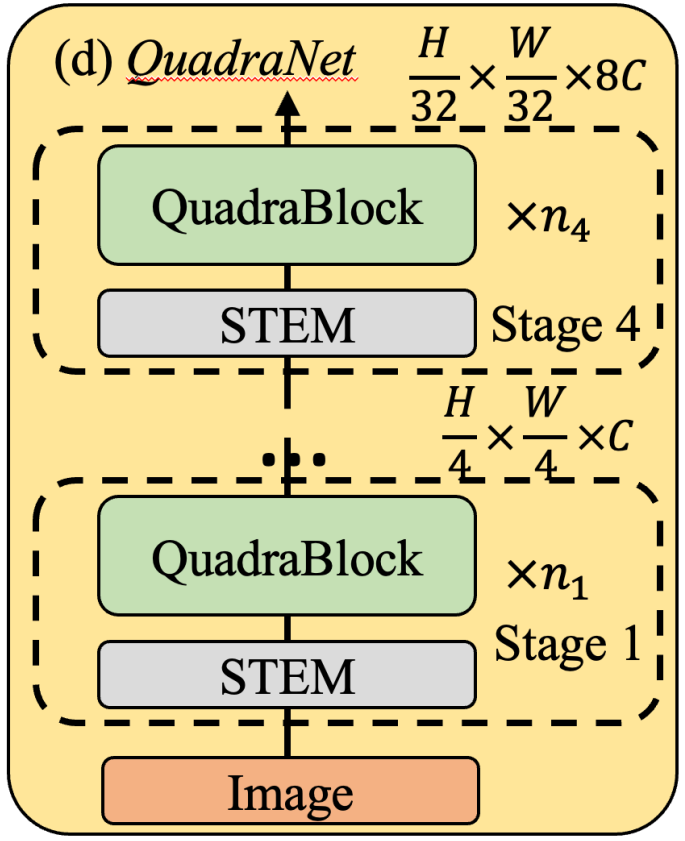
Pixel-wise spatial feature mixing
~5% of the parameters and MACs

Method: QuadraBlock

TABLE V: Channel-wise Quadratic Operator's Impact

Models	#Params	#FLOPs	Top-1 Acc.
QuadraNet36-T	23.6M	4.1G	82.2
+Quadratic 1*1 Conv1	44.6M	7.8G	82.3
+Quadratic 1*1 Conv2	65.6M	11.5G	82.7
QuadraNet36-B	90.4M	15.8G	84.1
+Quadratic 1*1 Conv1	174.3M	30.6G	83.9
+Quadratic 1*1 Conv2	258.2M	45.4G	84.2

Method: QuadraNet



- Following full-fledged 4-stage pyramid architecture
- Adjustable hyper-parameters:
  - Receptive field
  - Number of Channels
  - Number of Layers

Method: Hardware-aware QuadraNet Architecture Search

- Search Space: QuadraBlock and QuadraNet provide basic search units and search structures. The search space includes:
  - Kernel size (Receptive field)
  - Expansion coefficient (number of channels in each block)
  - Number of blocks in each stage

- Hardware Awareness:
  1. Analyzing the QuadraNet's memory and computing costs, and optimization factors.
  2. Projecting it into particular resource consumption estimations
  3. Establishing search feedback loop

- Search Method: ProxylessNAS (Han Cai, et al. in ICLR 2018)

Experiments: QuadraNet

TABLE I: Cognition and Computation Performance Evaluation

Model	#Para. (M)	FLOPs (G)	#Layer	Top-1 Acc. (%)	Throughput (img/s)
Swin-T[4]	29	4.5	12	81.3	1321
DeiT-S[17]	22	4.6	12	79.8	1621
TNT-S[18]	23.8	5.2	12	81.5	657
PVT-S[19]	24.5	3.8	16	79.8	1433
T2T-ViT-14[20]	21.5	4.8	14	81.5	1376
ConvNeXt-T[16]	28	4.5	18	82.1	1944
HorNet-T[9]	22	4	25	**82.8**	1254
QuadraNet25-T	16.2	2.9	25	81.2	**2433**
QuadraNet36-T	23.6	4.1	36	82.2	**1971**
Swin-S[4]	50	8.7	24	83.0	827
ConvNeXt-S[16]	50	8.7	36	83.1	1275
HorNet-S[9]	50	8.8	25	**83.8**	813
QuadraNet36-S	50.2	8.9	36	**83.8**	**1117**
Swin-B[4]	88	15.4	24	83.5	662
ConvNeXt-B[16]	89	15.4	36	83.8	969
HorNet-B[9]	87	15.6	25	**84.2**	641
QuadraNet25-B	61.2	11.1	25	83.6	**1138**
QuadraNet36-B	90.4	15.8	36	**84.1**	**892**

With competitive accuracy, QuadraNet achieves 40%-90% more throughput compared with Swin-Transformer (ICCV 2021) and HorNet (NeurIPS 2022)

Experiments: QuadraNet

- Less training time memory consumption

TABLE II: Memory Consumption Evaluation

Block	High-order	Traning Memory(MB) C=64	C=128
SkipBlock(No DW-Conv)	×	5515	12027
ConvBlock(Naive DW-Conv)	×	7364	15711
SwinBlock[4]	✓	12674	23472
HorBlock[9]	✓	10724	20715
QuadraBlock	✓	8108	17191
QuadraBlock*(w/ QuadraLib)	✓	7794	16730

Experiments: QuadraNet_NAS

We set the maximum latency for mobile CPU (CortexA76), VPU (Intel Myriad VPU), and GPU (A100) to 300ms, 30ms, and 3ms, respectively.

TABLE III: Hardware-Aware *QuadraNet* NAS

Searched Model	FLOPs	CPU	VPU	GPU	Top-1 Acc.
Proxyless-CPU	333M	299.6ms	53.1ms	1.8ms	72.4
Proxyless-VPU	275M	379.1ms	29.9ms	1.7ms	72.7
Proxyless-GPU	1065M	1521.1ms	146.1ms	2.9ms	74.4
QuadraNet_CPU	279M	299.8ms	34.7ms	2.1ms	73.1
QuadraNet_VPU	312M	437.1ms	29.9ms	1.9ms	73.2
QuadraNet_GPU	948M	1334.2ms	117.5ms	2.9ms	**76.7**

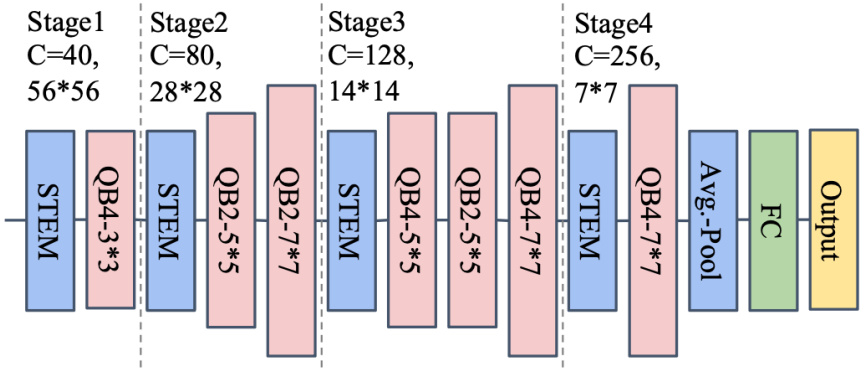


Fig. 4: Visualization of Searched *QuadraNet_VPU_10ms*

Conclusion

- Demystified the high-order neural interactions in Transformer-like neural networks through comprehensive analyses of algorithms and computation patterns and explored the bottleneck of existing optimization approaches.

- Presented *QuadraNet* -- a revolutionary neural network design methodology that incorporates efficient and powerful high-order neural interaction interactions.

- Generalized the quadratic neural networks design methodology to various model architectures and heterologous hardware computing constraints.

Q&A

Thank you!

Presenter: Chenhui Xu

Email: cxu21@gmu.edu