Low-complexity Fixed-point Convolutional Neural Networks for Automatic Target Recognition

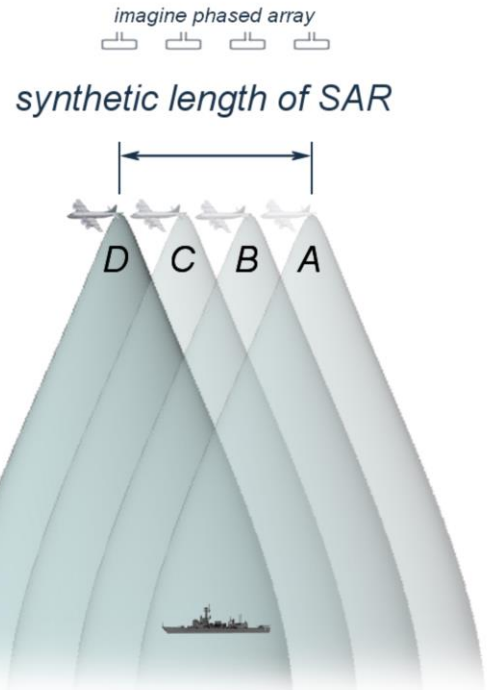
Hassan Dbouk*, *Hanfei Geng*, Craig M. Vineyard**, and Naresh R. Shanbhag**

*Dept. of Electrical and Computer Engineering, University of Illinois at Urbana Champaign

**Sandia National Laboratories

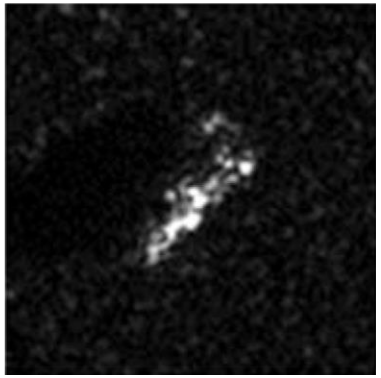
ICASSP2020
Barcelona

Automatic Target Recognition (ATR)



- ATR has been an active area of research for decades
  - synthetic aperture radar (SAR) imagery guarantees robust operation
- Deployed on resource-constrained airborne vehicles
  - real-time and always-on detection of targets is required
- Accuracy of ATR systems cannot be compromised
  - deep learning-based solutions have gained momentum

SAR image



Prior Art: Deep Networks for ATR

Network Architecture	Number of Parameters	Number of MACs	Best Reported Accuracy [%]
Morgan [1]	88K	25M	92.3
Wagner [2]	410K	10M	99.5
Gao [3]	115K	6M	97.8
Ding [4]	231M	2B	93.2
Chen [5]	303K	42M	99.1

- Existing works focus on achieving the best classification accuracy
  - ignore the cost of implementing these networks
- The models require floating-point arithmetic for implementation
  - prohibitive on resource-constrained devices

Contributions

- We present the design of low-complexity networks for ATR with minimal loss in classification accuracy via:
  - compact network architecture design
  - training networks with reduced precision activations and weights

- Our proposed networks achieve a **total $\mathbf{984\times}$ reduction** in representational cost and **$\mathbf{71\times}$ reduction** in computational cost compared to the best CNN in the SAR ATR literature
  - while achieving $> 99\%$ classification accuracy on the MSTAR dataset

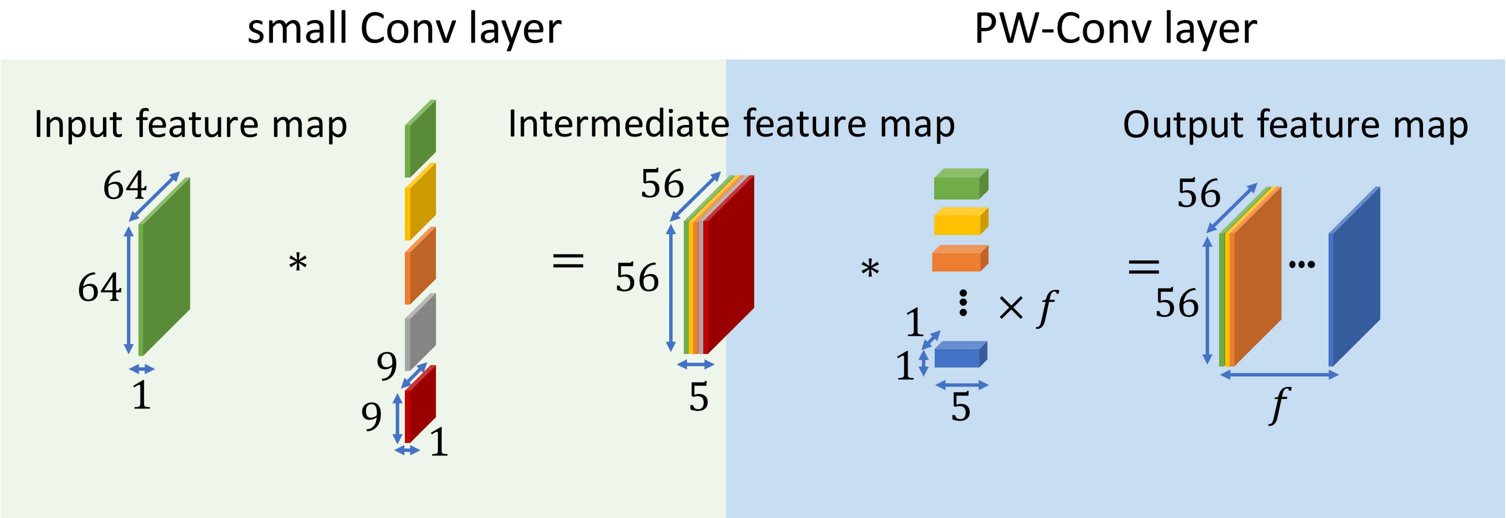
Compact Network Architecture

- Parameterizable by f
  - controls the width of the network (complexity)
- 1st layer typically dominates complexity
  - standard 3D convolution will contribute to 99% of network complexity
- BatchNorm (BN) layers allow for training smaller models for the same accuracy
  - learning is easier when input statistics are normalized

Layer Type	Layer Shape	Input Shape
Conv	$9 \times 9 \times 1 \times 5$	$64 \times 64 \times 1$
BN	5	$56 \times 56 \times 5$
ReLU	–	$56 \times 56 \times 5$
PW-Conv	$1 \times 1 \times 5 \times f$	$56 \times 56 \times 5$
BN	f	$56 \times 56 \times f$
ReLU	–	$56 \times 56 \times f$
MaxPool	8×8	$56 \times 56 \times f$
Conv	$2 \times 2 \times f \times 2f$	$7 \times 7 \times f$
BN	$2f$	$6 \times 6 \times 2f$
ReLU	–	$6 \times 6 \times 2f$
MaxPool	2×2	$6 \times 6 \times 2f$
Conv	$2 \times 2 \times 2f \times 4f$	$3 \times 3 \times 2f$
BN	$4f$	$2 \times 2 \times 4f$
ReLU	–	$2 \times 2 \times 4f$
Conv	$2 \times 2 \times 4f \times 10$	$2 \times 2 \times 4f$
BN	10	$1 \times 1 \times 10$
ReLU	–	$1 \times 1 \times 10$
FC	10×10	$1 \times 1 \times 10$
Softmax	–	$1 \times 1 \times 10$

Compact Network Architecture – 1st Layer

- Factorize the 1st layer into two layers:
  - small convolution layer (5 kernels instead of f)
  - pointwise convolution layer



complexity reduction of 2.6 × - 4.6 ×

Layer Type	Layer Shape	Input Shape
Conv	$9 \times 9 \times 1 \times 5$	$64 \times 64 \times 1$
BN	5	$56 \times 56 \times 5$
ReLU	–	$56 \times 56 \times 5$
PW-Conv	$1 \times 1 \times 5 \times f$	$56 \times 56 \times 5$
BN	f	$56 \times 56 \times f$
ReLU	–	$56 \times 56 \times f$
MaxPool	8×8	$56 \times 56 \times f$
Conv	$2 \times 2 \times f \times 2f$	$7 \times 7 \times f$
BN	$2f$	$6 \times 6 \times 2f$
ReLU	–	$6 \times 6 \times 2f$
MaxPool	2×2	$6 \times 6 \times 2f$
Conv	$2 \times 2 \times 2f \times 4f$	$3 \times 3 \times 2f$
BN	$4f$	$2 \times 2 \times 4f$
ReLU	–	$2 \times 2 \times 4f$
Conv	$2 \times 2 \times 4f \times 10$	$2 \times 2 \times 4f$
BN	10	$1 \times 1 \times 10$
ReLU	–	$1 \times 1 \times 10$
FC	10×10	$1 \times 1 \times 10$
Softmax	–	$1 \times 1 \times 10$

Training Fixed-Point Networks

- Quantize both weights and activations in the forward path
    - keep full-precision copies of the weights for weight updates

- Two key challenges:
    - determining a suitable clipping value for quantization
    - back-propagating the gradients through non-differentiable quantization function

Training Fixed-Point Networks – Clipping

- Weights clipping:

$$c_{W,l} = \max(|W_l|)$$

- Activations clipping:

$$c_{A,l} = \max_{i \in [C_l]} \left(\beta_l^{(i)} + 3\gamma_l^{(i)}\right)$$

guarantees
$\Pr\{x_l \leq c_{A,l}\} \geq 0.99865$

- Where for every layer $l \in \{1,2, \dots, L\}$:
  - $|.|$ is the element-wise absolute value operator
  - C_l is the number of channels in the input activation tensor
  - $(\beta_l^{(i)}, \gamma_l^{(i)})$ are the learnable per-channel shift and scale BN parameters

Training Fixed-Point Networks – STE

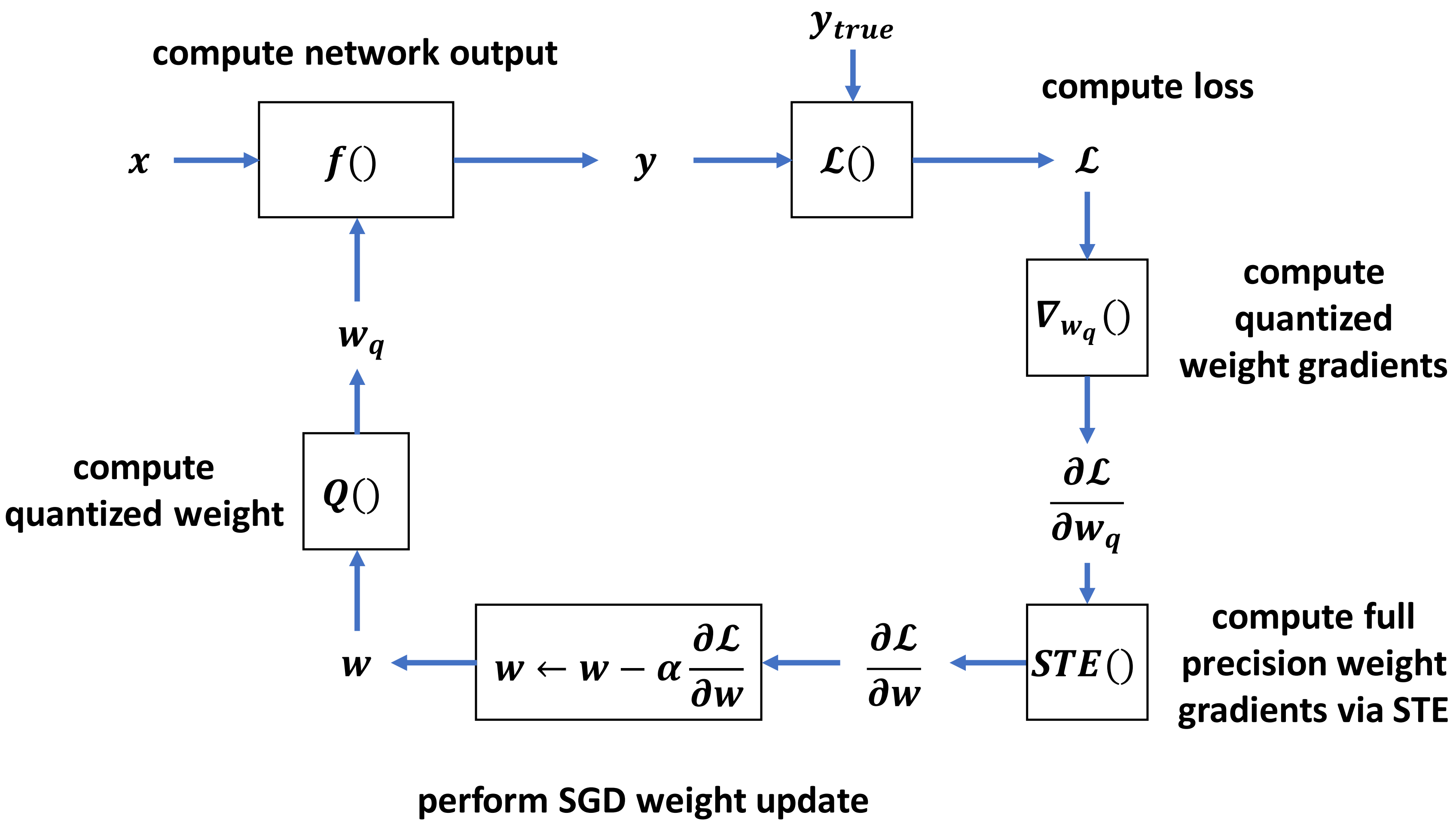
- Use the straight-through estimator (STE) for calculating the gradients of the quantization function:

[Bengio - arXiv'13]

$$\frac{\partial \mathcal{L}}{\partial x} = \frac{\partial \mathcal{L}}{\partial x_q} \times \frac{\partial x_q}{\partial x} \approx \frac{\partial \mathcal{L}}{\partial x_q} \times \mathbb{I}\{c_1 \leq x \leq c_2\}$$

- $x_q = Q(x)$ is the quantized signal
- c_1, c_2 are the quantizer clipping values
- $\mathcal{L}$ is the loss function

Training Fixed-Point Networks – Methodology



Complexity Metrics – Computational Cost

- Captures the number of 1-b full adders (FA) needed to implement the multiplications required for a single inference

$$\mathcal{C}_C = \sum_{l=1}^{L} N_l D_l B_{W,l} B_{A,l}$$

- Where for every layer $l \in \{1,2,\ldots,L\}$ we have:
  - N_l is the number of dot products
  - D_l is the dot product dimensionality
  - $B_{W,l}$ and $B_{A,l}$ are the weights and activations bit precisions respectively

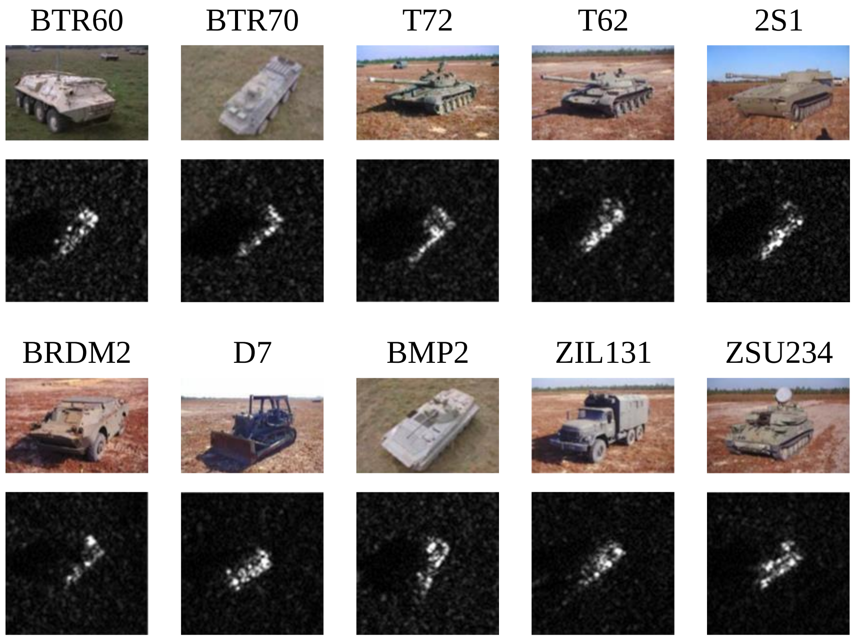
Complexity Metrics – Representational Cost

- Measures the number of bits needed to represent the entire network for a single inference:

$$\mathcal{C}_R = \sum_{l=1}^{L} \left(|W_l| B_{W,l} + |A_l| B_{A,l} \right)$$

- Where for every layer $l \in \{1,2, \dots, L\}$ we have:
  - $|W_l|$ and $|A_l|$ are the number of elements in the weights and activations tensors respectively
  - $B_{W,l}$ and $B_{A,l}$ are the weights and activations bit precisions respectively

Experimental Setup – MSTAR Dataset



Vehicle Type	Training Images (17 degrees)	Testing Images (15 degrees)
2S1	299	274
BMP2	698	587
BRDM2	298	274
BTR60	256	195
BTR70	233	196
D7	299	274
T62	299	273
T72	691	582
ZIL131	299	274
ZSU234	299	274

- Benchmark our networks using the publicly available MSTAR dataset
  - standard dataset for SAR-based ATR systems

Floating-Point Results – Accuracy

for a fair comparison, all the models were trained using the **same hyperparameter setup**

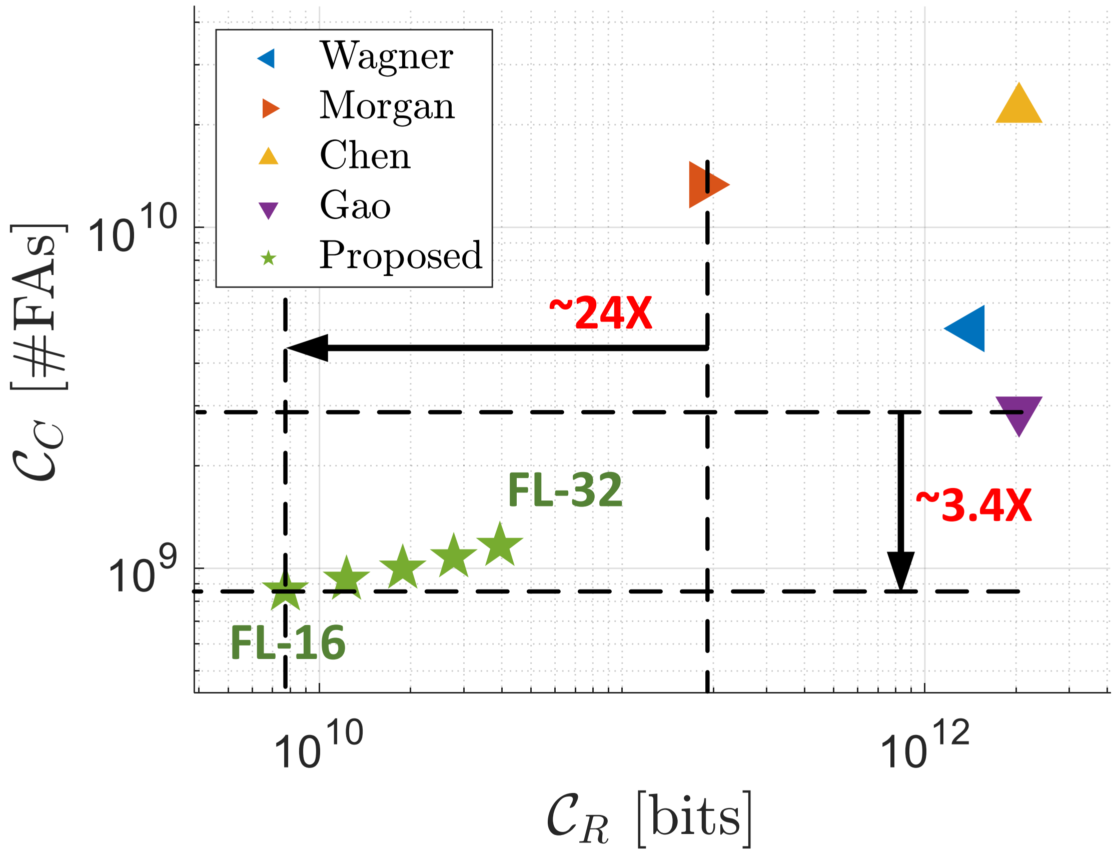
- Comparing the classification accuracy of our proposed networks with existing network topologies
  - proposed low-complexity networks remain competitive with $> 99\%$ accuracy
- FL-x denotes our proposed floating-point network with $f = x$
  - increasing f improves performance

Network Architecture	Input Crop Size	Test Accuracy [%]
Prior Art		
Morgan [1]	128×128	99.72
Wagner [2]	64×64	99.56
Gao [3]	64×64	99.31
Ding [4]	128×128	99.34
Chen [5]	88×88	99.66
Proposed Networks		
FL-16	64×64	99.38
FL-20	64×64	99.47
FL-24	64×64	99.41
FL-28	64×64	99.56
FL-32	64×64	99.66

Floating-Point Results – Complexity

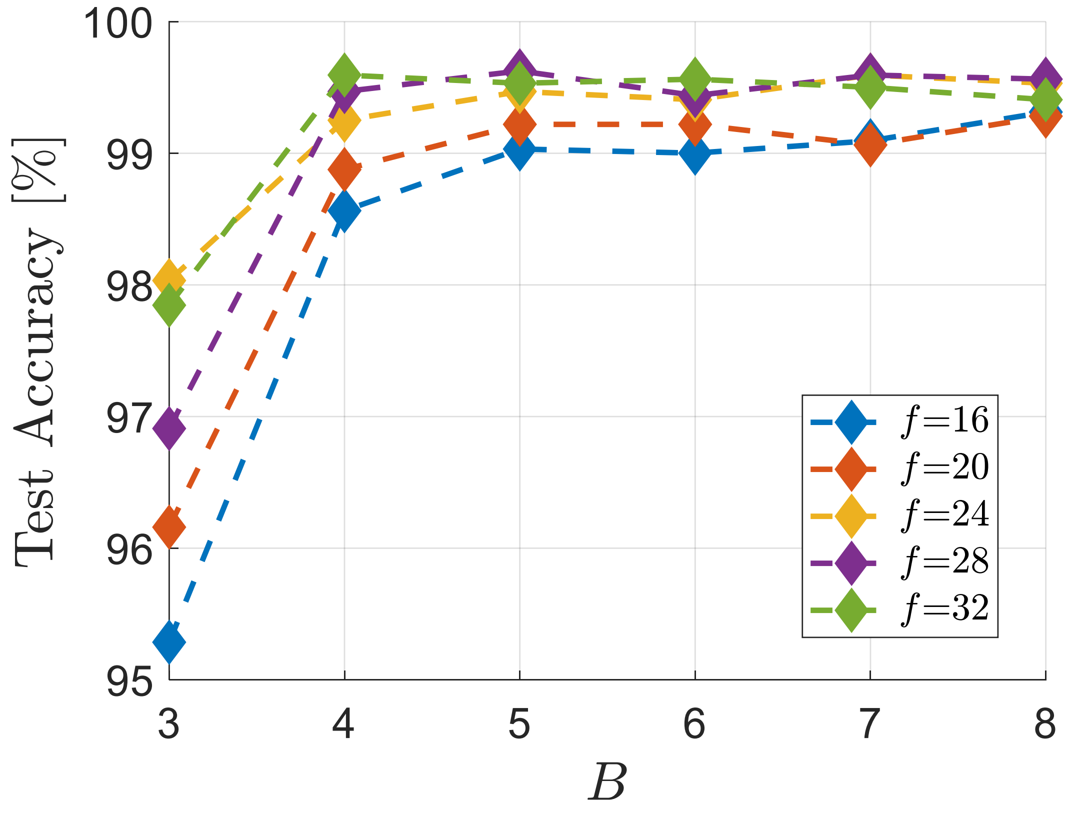
- **At iso-accuracy**, our proposed networks achieve massive reductions in complexity
  - increasing f increases the complexity

- FL-16 achieves **$24\times$ reduction** in $\mathcal{C}_R$ and **$3.4\times$ reduction** in $\mathcal{C}_C$



Fixed-Point Results – Impact of Bit Precision

- Fix the weight and activation precision $B_{W,l} = B_{A,l} = B \; \forall l \in [L]$
  - simplifies the search space
- Using 4bits is sufficient to achieve $>$ 99% accuracy
  - massive reductions compared to 32b floating-point

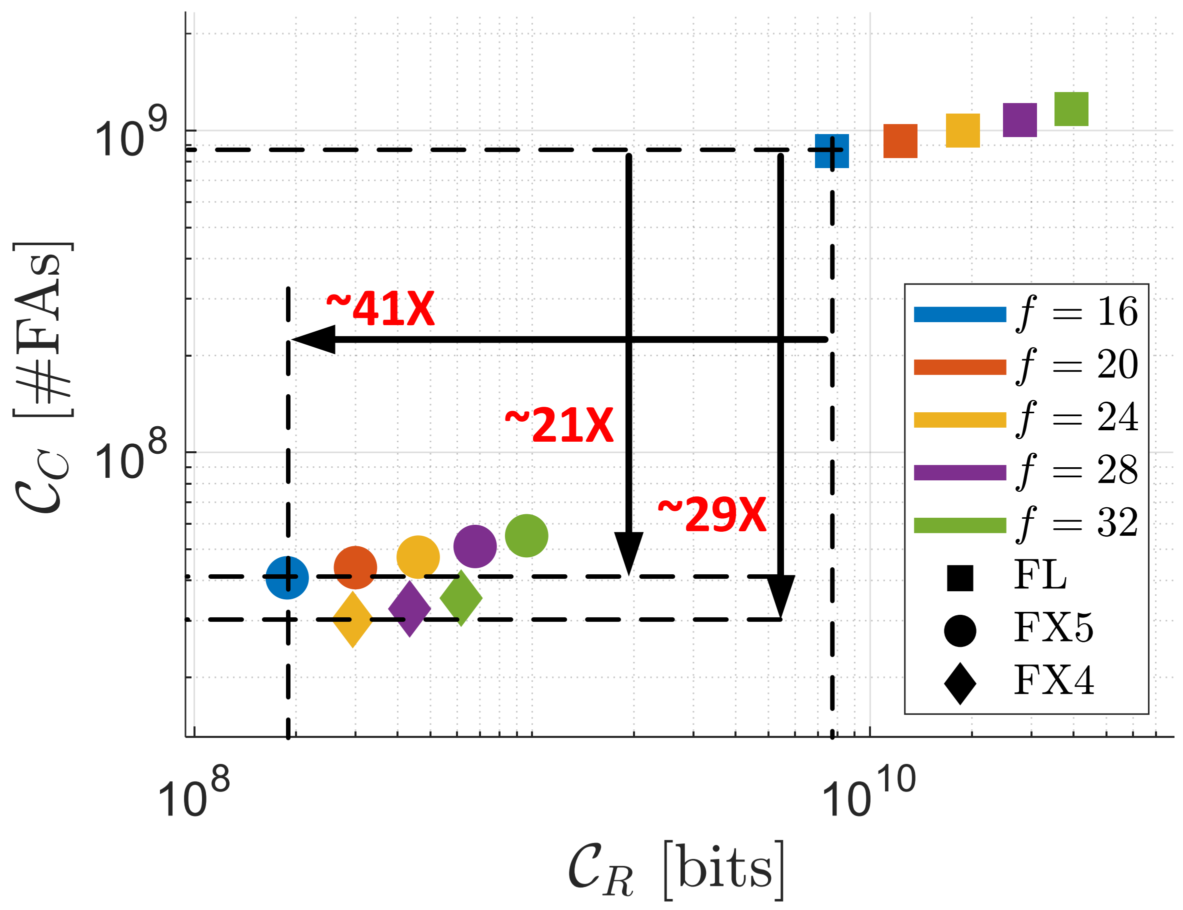


Fixed-Point Results – Comparison

- **At iso-accuracy**, our proposed fixed-point (FX) networks achieve massive reductions in complexity

- FX5-16 achieves **41 × reduction** in $\mathcal{C}_R$ and **21 × reduction** in $\mathcal{C}_C$

All models achieving > 99% accuracy



Conclusion & Future Work

- We have presented a set of compact CNN architectures for ATR coupled with a fixed-point training methodology
- The proposed networks achieve a total **$984 \times$ reduction** in $\mathcal{C}_R$ and **$71 \times$ reduction in $\mathcal{C}_C$** compared to SOTA CNNs for ATR, at **iso-accuracy ($> 99\%$)** on the MSTAR dataset
- Future work: mapping the proposed networks onto efficient hardware architectures to further facilitate their deployment

Thank you!

Acknowledgement:

This work is supported by a grant from Sandia National Laboratories and DARPA FRANC Program.