Semiconductor Research Corporation

CoCoSys
CENTER FOR THE CO-DESIGN OF COGNITIVE SYSTEMS

Georgia Tech

SYNERGY

Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference

Presenter: Akshat Ramachandran (**Advisor:** Tushar Krishna)

Collaborators: Zishen Wan, Geonhwa Jeong, John Gustafson (ASU), Souvik Kundu (Intel)

Georgia Institute of Technology

Contact: akshat.r@gatech.edu (Akshat Ramachandran)

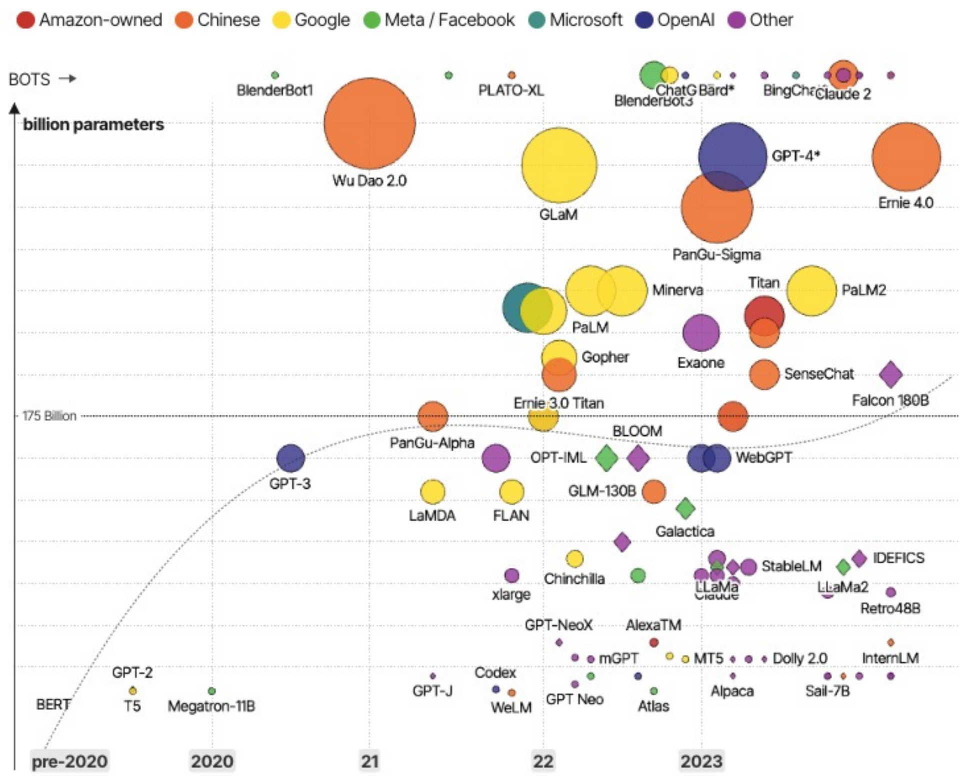
SRC

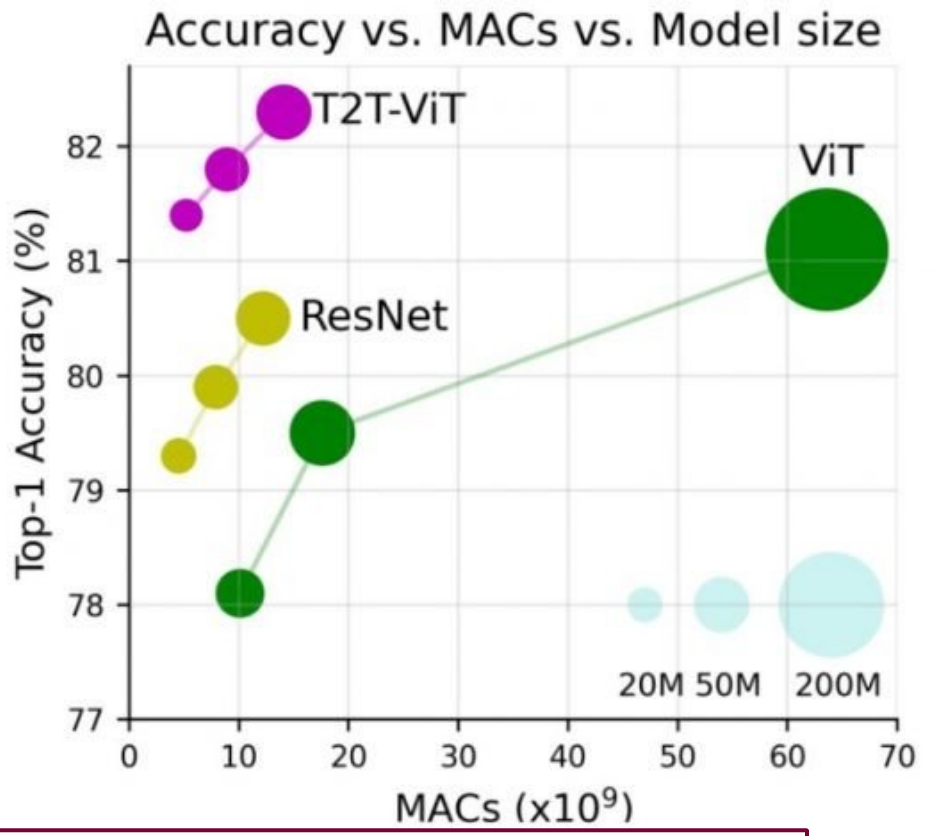
May 29, 2024

Executive Summary

- Why ?
  - Existing quantization data-formats lack adaptability and/or are inefficient for implementation in resource-constrained devices.
  - Lack of a generalized quantization framework in prior art.
- What ?
  - Design of a composite data type called Logarithmic Posits (LP) that blends the adaptability of posits with hardware efficiency of Logarithmic Numbers.
  - Novel genetic-algorithm based quantization framework leveraging contrastive learning for identifying layer-wise mixed-precision quantization parameters.
  - Unified mixed-precision Logarithmic Posit Accelerator (LPA) for high throughput DNN inference.
- How ?
  - Employs a co-design approach, integrating the development of the LP data type, the quantization framework, and the LPA architecture. This integrated approach allows for dynamic adaptation to DNN parameter distributions, resulting in highly efficient DNN inference.

From Bulky DNNs to Sleek Edge Deployment!





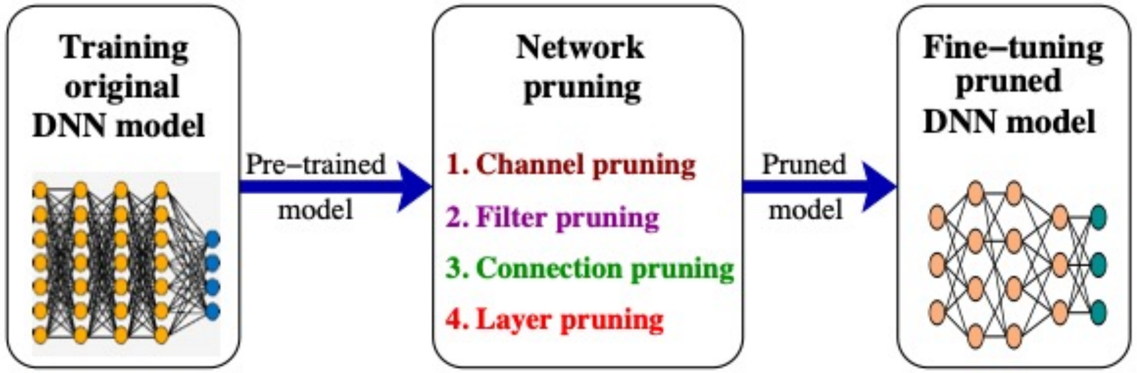
- ❖ YoY increase in DNN sizes leads to escalating computational and storage demands!
- ❖ Limited compute, storage resources and energy budget of edge devices (e.g., phones) makes deployment challenging!

SRC

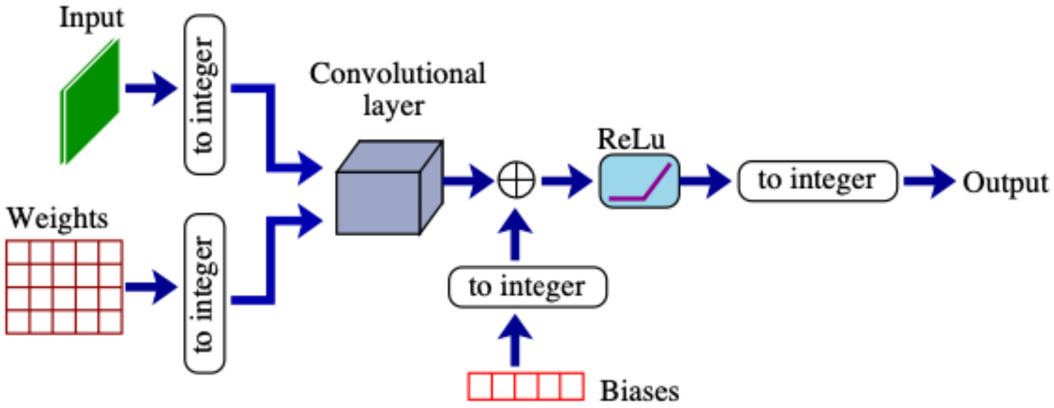
Source: NVIDIA, Viso.ai, LifeArchitect.ai

From Bulky DNNs to Sleek Edge Deployment!

Model Compression techniques for efficient DNN deployment:



Pruning

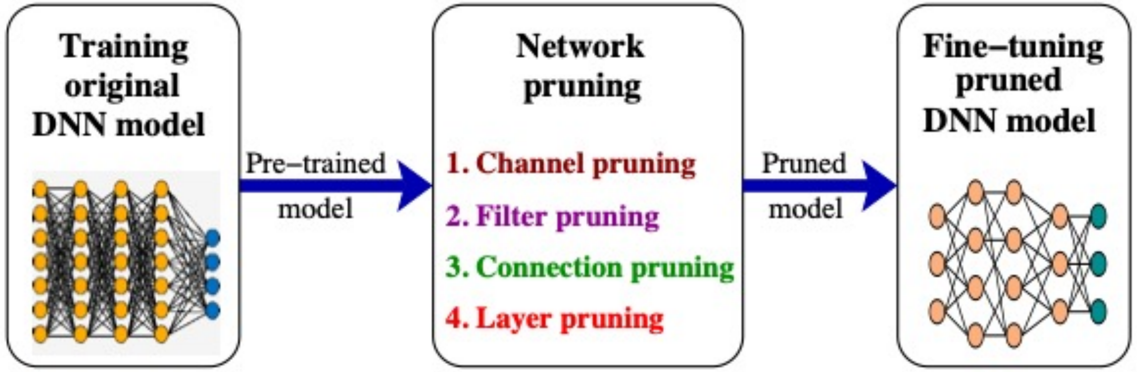


Quantization

Source: Rahul, M et al. "A survey on deep neural network compression: challenges, overview, and solutions." arXiv preprint arXiv:2010.03954 (2020).

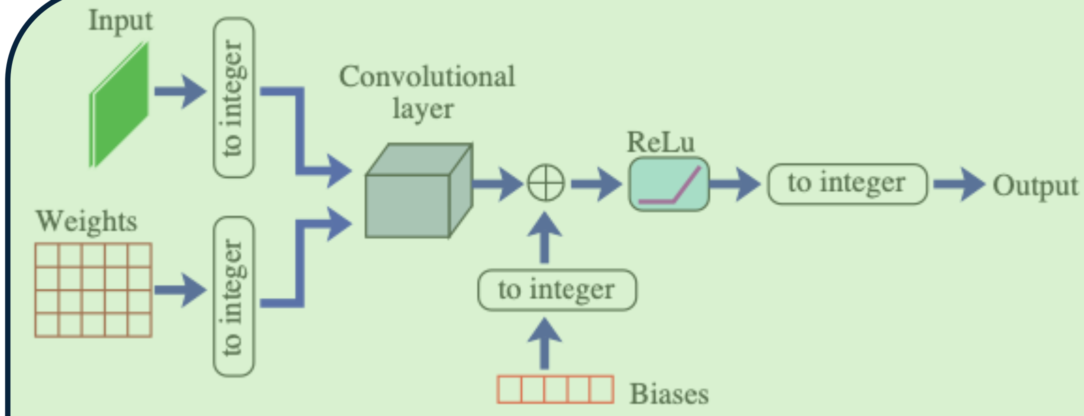
From Bulky DNNs to Sleek Edge Deployment!

Model Compression techniques for efficient DNN deployment:



Pruning

Focus of our work!



Quantization

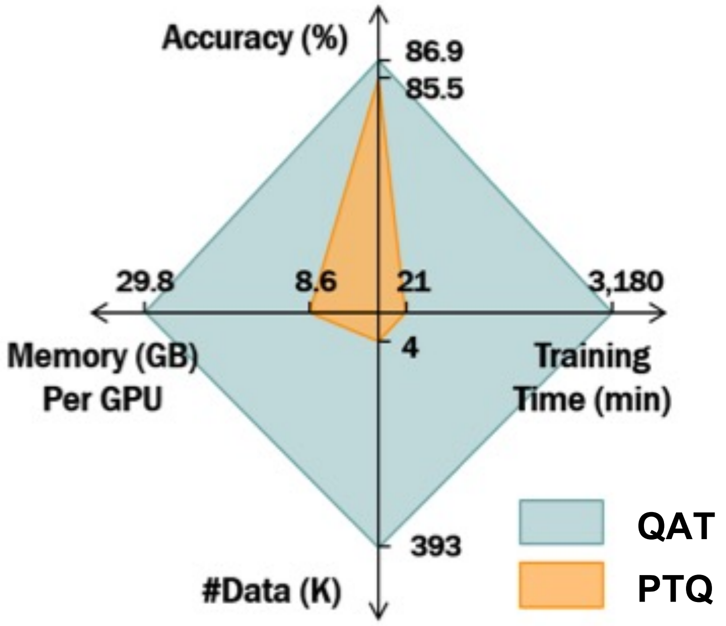
Source: Rahul, M et al. "A survey on deep neural network compression: challenges, overview, and solutions." arXiv preprint arXiv:2010.03954 (2020).

Background: Types of Quantization Techniques

Quantization Aware Training (QAT):

Higher Accuracy
Improved Model Robustness

❌ Increased training complexity and time
❌ Large-scale data dependency



Focus of our work!

Post Training Quantization (PTQ):

Speed and Simplicity
Low data requirement (Can also be data-free)

❌ Potential accuracy drop
❌ Sensitivity to calibration dataset

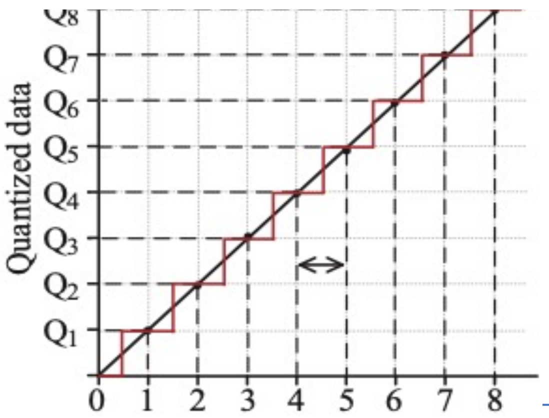
Source: Bai, Haoli, et al. "Towards efficient post-training quantization of pre-trained language models." NeurIPS 2022.

Background: Types of Quantization Formats

Uniform Quantization

- Assigns same quantization step size cross the entire range of values.
- Results in simpler hardware but leads to higher quantization error for distributions with large variances.
- Example: Integers.

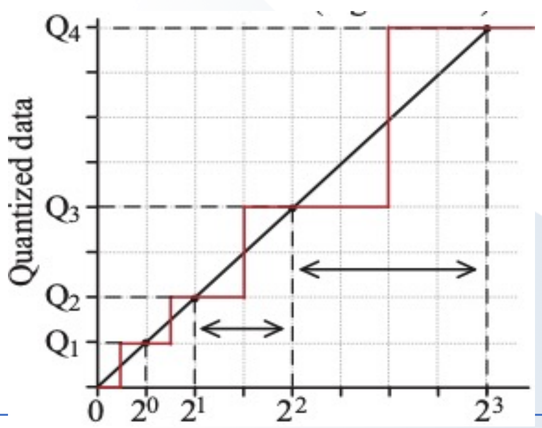
$$Q(X,\gamma,b) = clip(\left\lfloor \frac{X}{\gamma} \right\rceil, -2^{b-1}+1, 2^{b-1}-1)$$



Non-Uniform Quantization

- Has variable distribution intervals, for selective quantization for significant data or extensive dynamic ranges.
- Potentially complex hardware and requires automated quantization algorithms.
- Example: IEEE-754 Floating-Point, Posits, Vector Quantization, Logarithmic Posits.

$$Q(X,b,e,f) = sign(X) \times 2^{e} \times 1.f$$



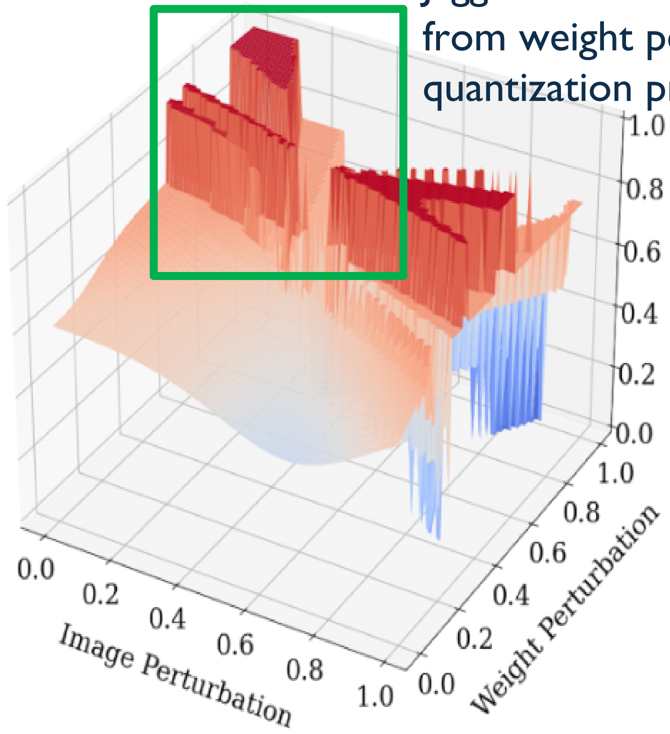
Source: Liu, Fangxin, et al. "Improving neural network efficiency via post-training quantization with adaptive floating-point." CVPR. 2021.

SRC

Problem: Non-smooth Loss Landscape

DNNs particularly depict a non-smooth quantization loss landscape.

Jagged and non-smooth peaks arise from weight perturbations during the quantization process

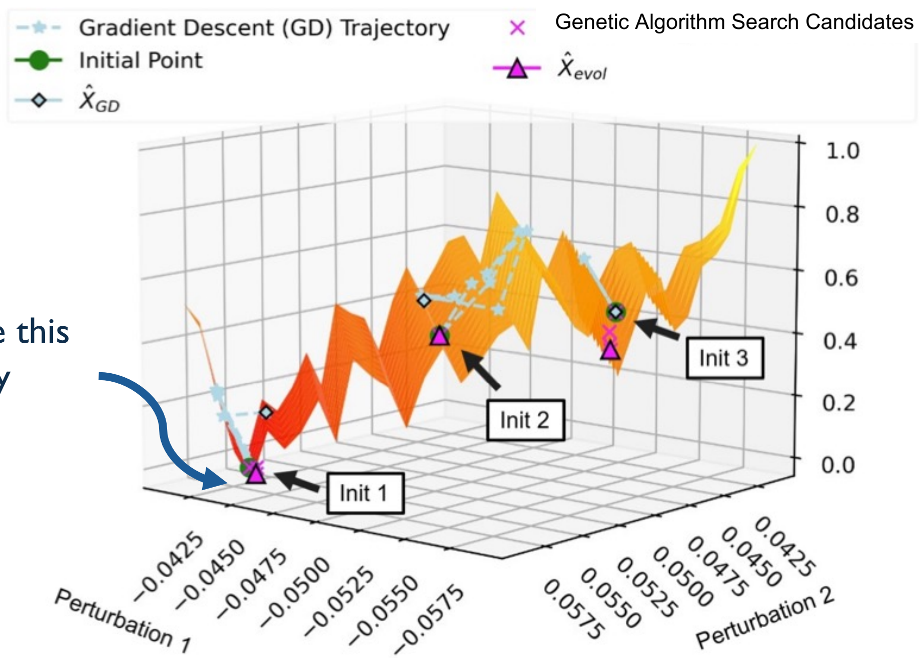


Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

Problem: Gradient-Descent Cannot Traverse this Landscape

First and second-order gradient cannot be used satisfactorily to traverse this non-smooth loss landscape because of the presence of multiple local-minima.

GA-based search is able to traverse this loss landscape and optimally identify quantization parameters



SRC

Our Solution

We therefore adopt a GA based layer-wise quantization framework.

Why layer wise ? Perturbing too many layers at a time can cause traversal of a high-dimensional search space and does not guarantee convergence.

SRC

Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

Background: Genetic Algorithm

Consider a population of randomly rabbits: some individuals are potentially faster and smarter than others.

Background: Genetic Algorithm

- Slower, dumber rabbits are likely to be caught and eaten by foxes.
- Fast, smart rabbits survive to do what rabbits to best: make more rabbits!!

SRC

Background: Genetic Algorithm

- The rabbits that survive breed with each other to generate offspring, which starts to mix up their genetic traits
- – Fast rabbits might breed with fast rabbits
- – Fast rabbits with slow rabbits
- – Smart with not-so-smart, etc…

- Furthermore, nature occasionally throws in a “wild hare” because genes can mutate.

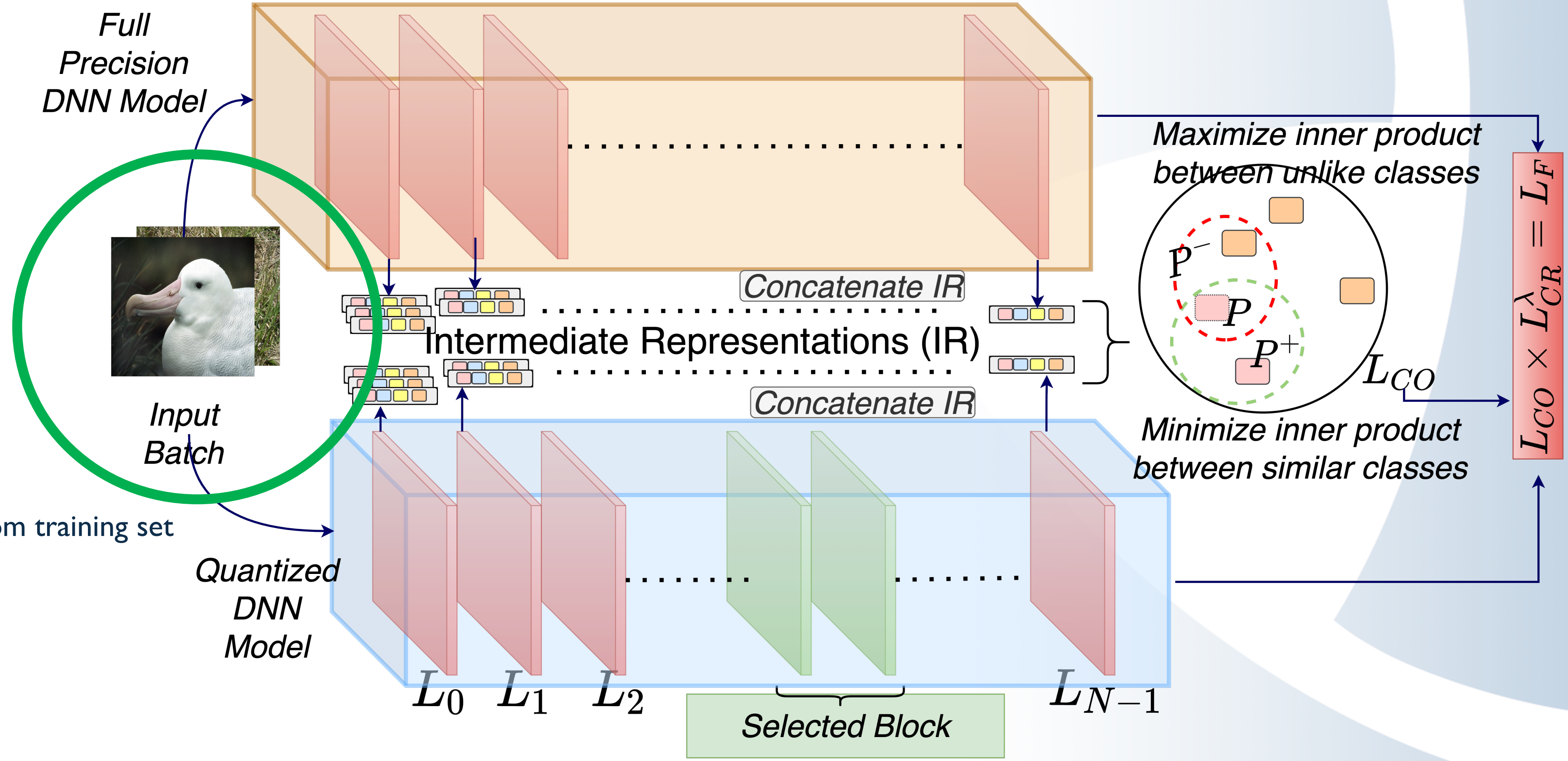
SRC

Background: Genetic Algorithm

- At the end the rabbits that survive will be the fastest, strongest and smartest among the population because the fox would've eaten the slow, weak and not-so-smart ones.

- In this analogy, an individual rabbit represents a solution to the problem (i.e. a single point in the state space).
- The foxes represent the problem constraints – Solutions that do well are likely to survive.
- We create similar such notions for quantization.

Proposed: Automated Quantization Framework

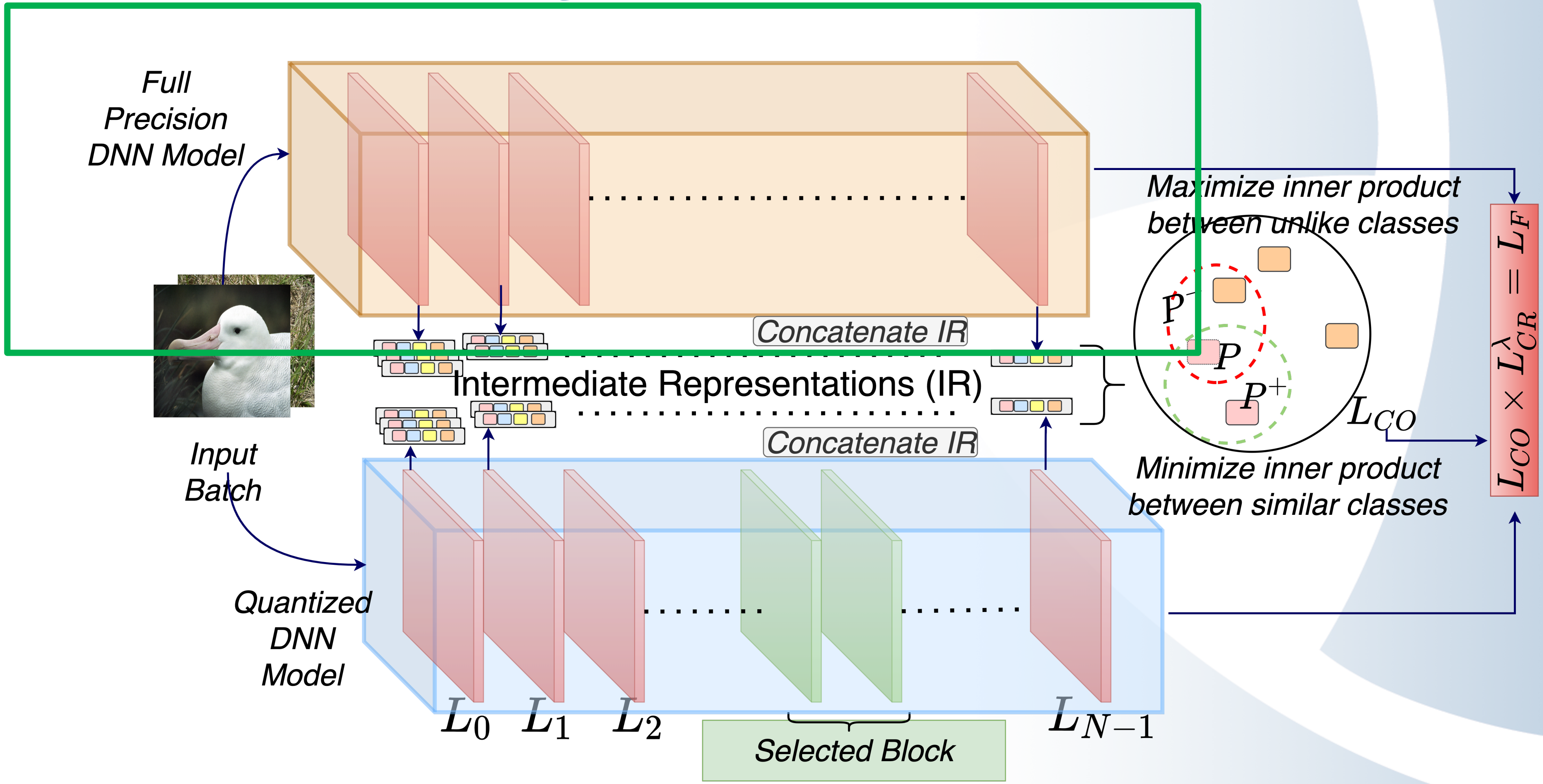


128 calibration images drawn from training set

Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

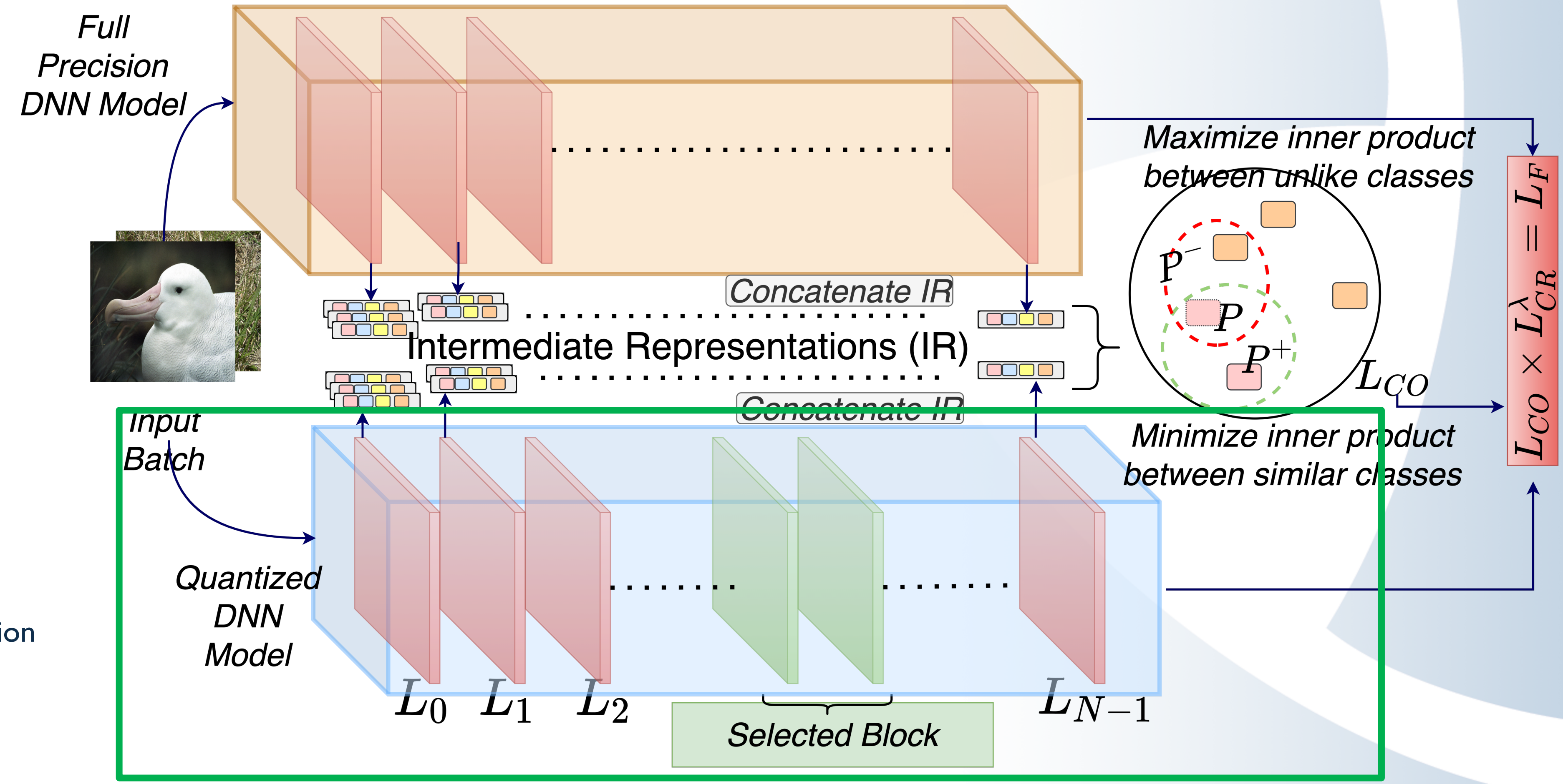
Proposed: Automated Quantization Framework

Full-Precision Model



Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

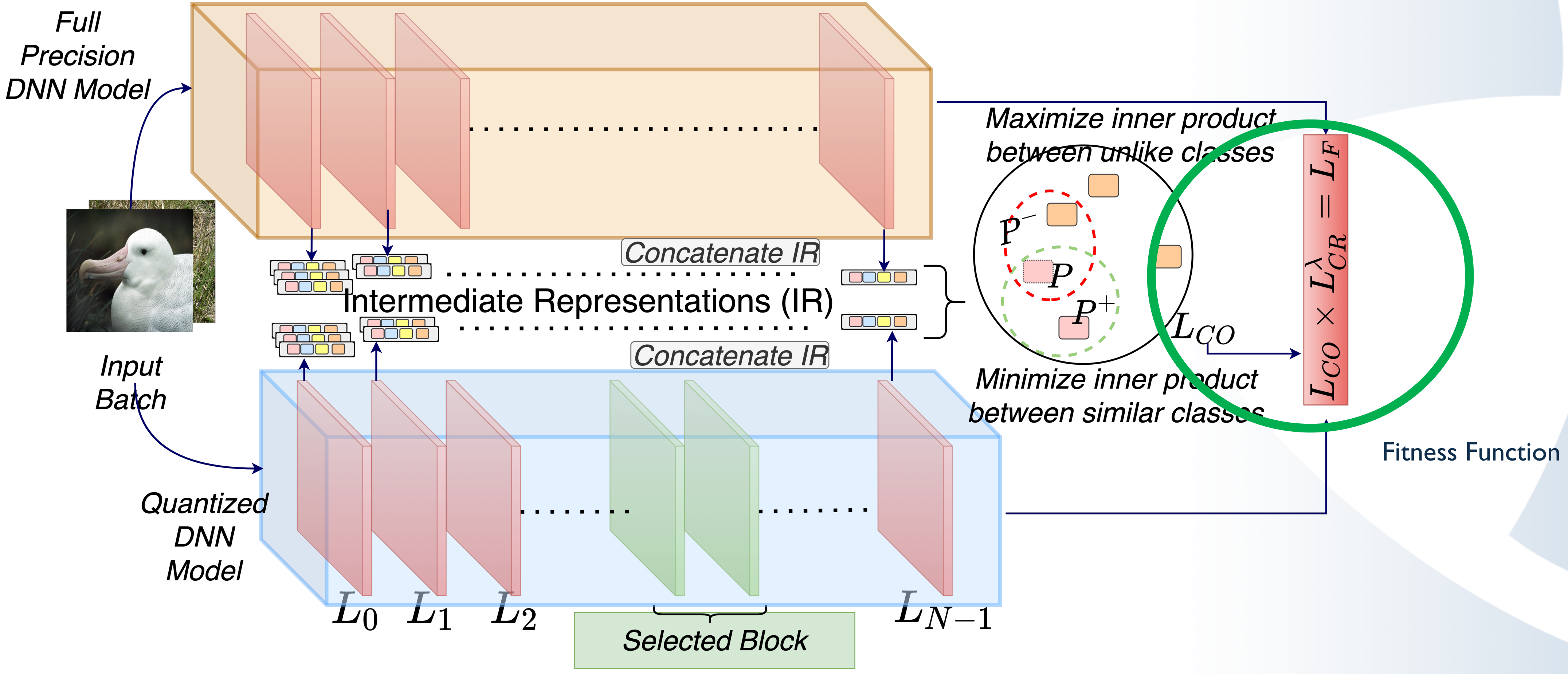
Proposed: Automated Quantization Framework



Randomly initialized model for quantization

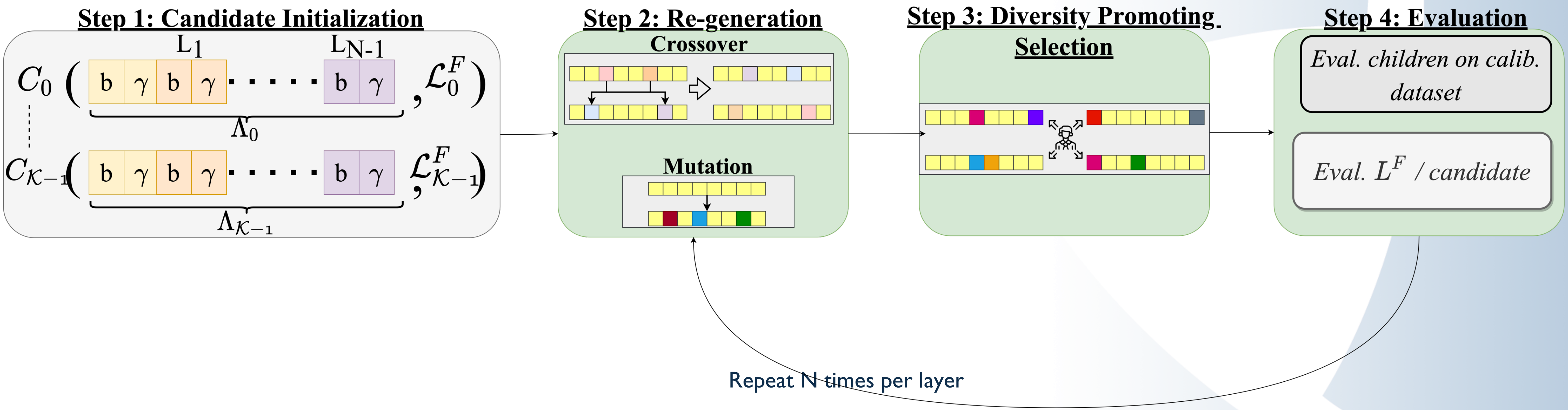
Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

Proposed: Automated Quantization Framework



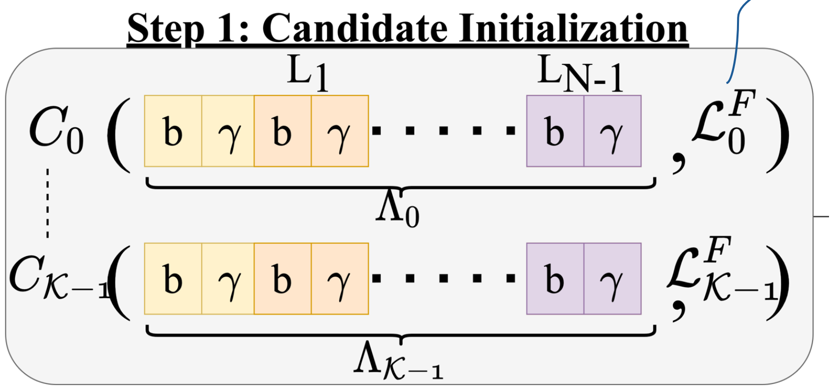
Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

Proposed: Automated Quantization Framework



Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

Automated Quantization Framework: Step 1



Fitness value of each candidate

A quantization solution comprises an encoded vector Δ of length $2N$ and each set of 2 values represent the 2 integer quantization parameters of a layer l.

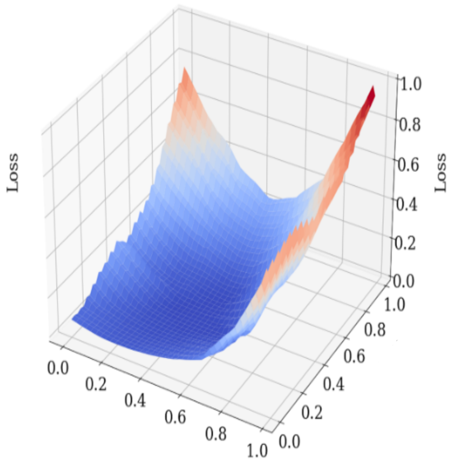
$$Q(X, \gamma, b) = clip(\left\lfloor \frac{X}{\gamma} \right\rceil, -2^{b-1} + 1, 2^{b-1} - 1)$$

Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

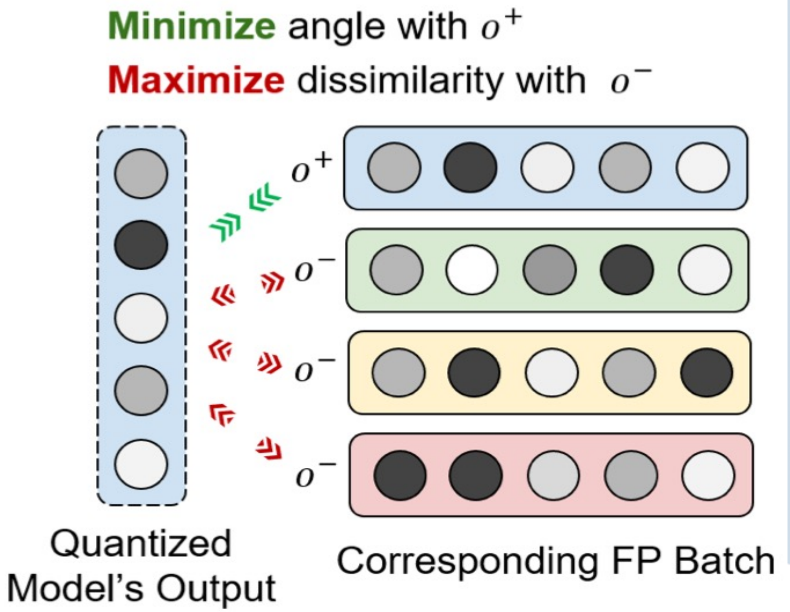
Automated Quantization Framework: Fitness Function Calculation

- Experiments in self-supervised learning and our own experiments suggest that contrastive learning tend to smooth the loss landscape.

$$\mathcal{L}^{C}_{i,j} = -\log \frac{\sum_{p+} \exp(\lambda^{p}_{i,j} \cdot \lambda^{p+}_{i,j} / \tau)}{\sum_{p+} \exp(\lambda^{p}_{i,j} \cdot \lambda^{p+}_{i,j} / \tau) + \sum_{p-} \exp(\lambda^{p}_{i,j} \cdot \lambda^{p-}_{i,j} / \tau)}$$

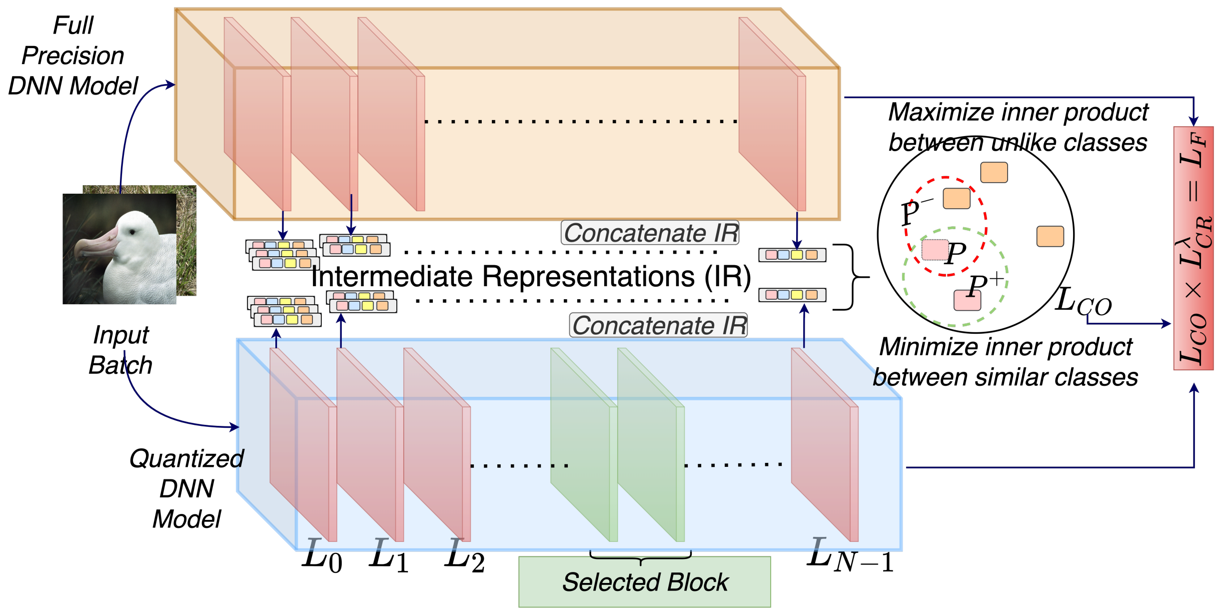


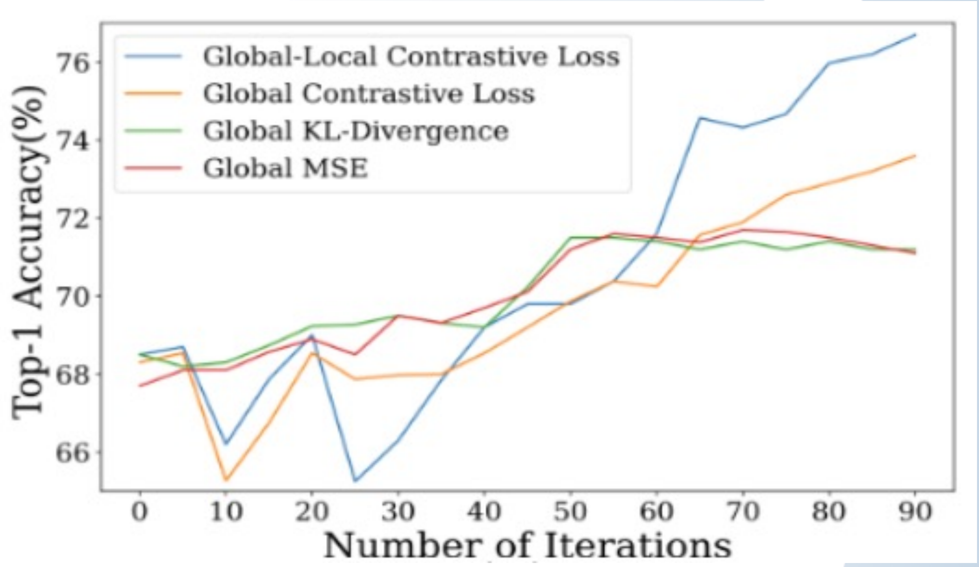
With Contrastive Learning



Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

Automated Quantization Framework: Fitness Function Calculation



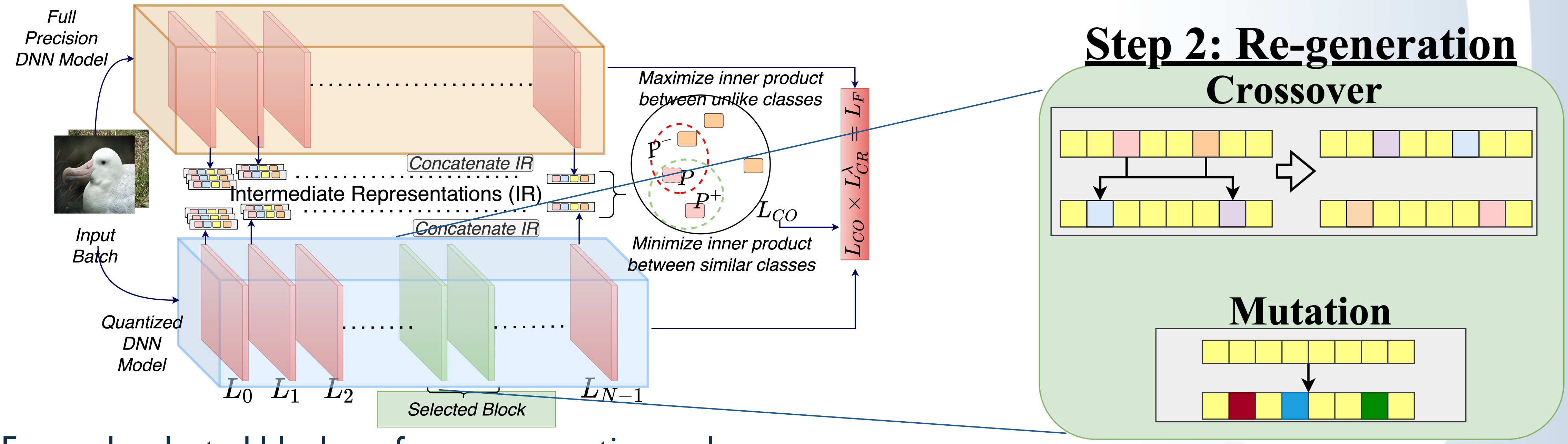


$$\mathcal{L}_{\text{CR}} = \sum_{l \in N} \#\text{PARAM}(\mathbf{H}_l^{\text{FP}}) \times n_l$$

The contrastive component of fitness function aims to align the distribution of quantized model's intermediate representations closely with the FP model, while the compression ratio metric incentivizes lower bit-widths

Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

Automated Quantization Framework: Step 2



For each selected block perform regeneration and crossover.

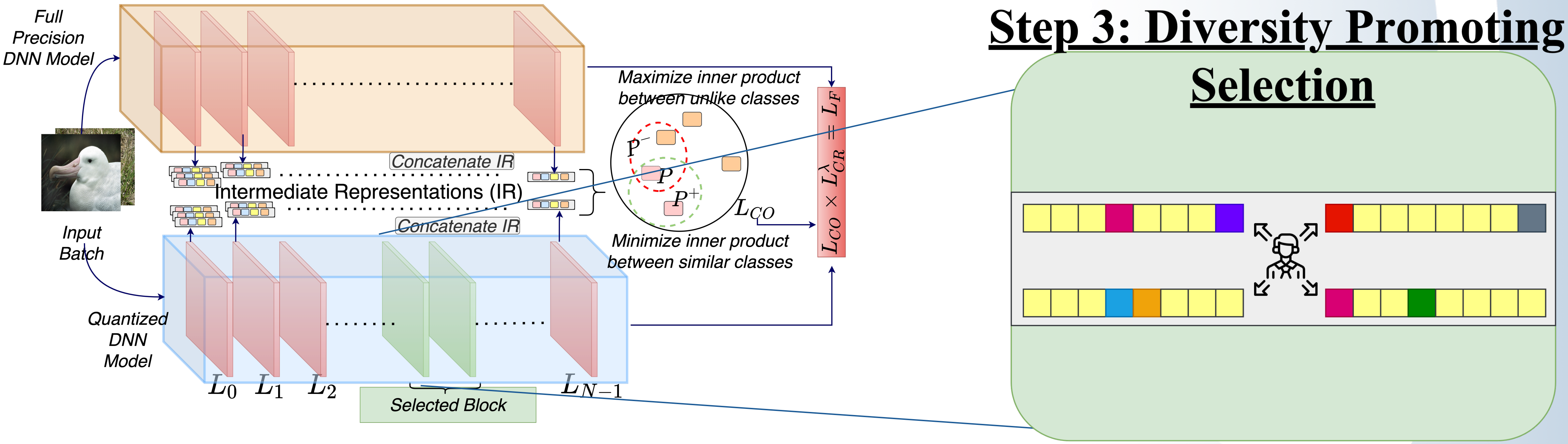
$$b_{child} = \mathbf{random}(\mathbf{min}(b_{p1}, b_{p2}) - 1, \mathbf{max}(b_{p1}, b_{p2}) + 1)$$

$$\gamma_{child} = \mathbf{mean}(\gamma_{p1}, \gamma_{p2}) + \eta(-10^{-3}, 10^{3})$$

SRC

Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

Automated Quantization Framework: Step 3

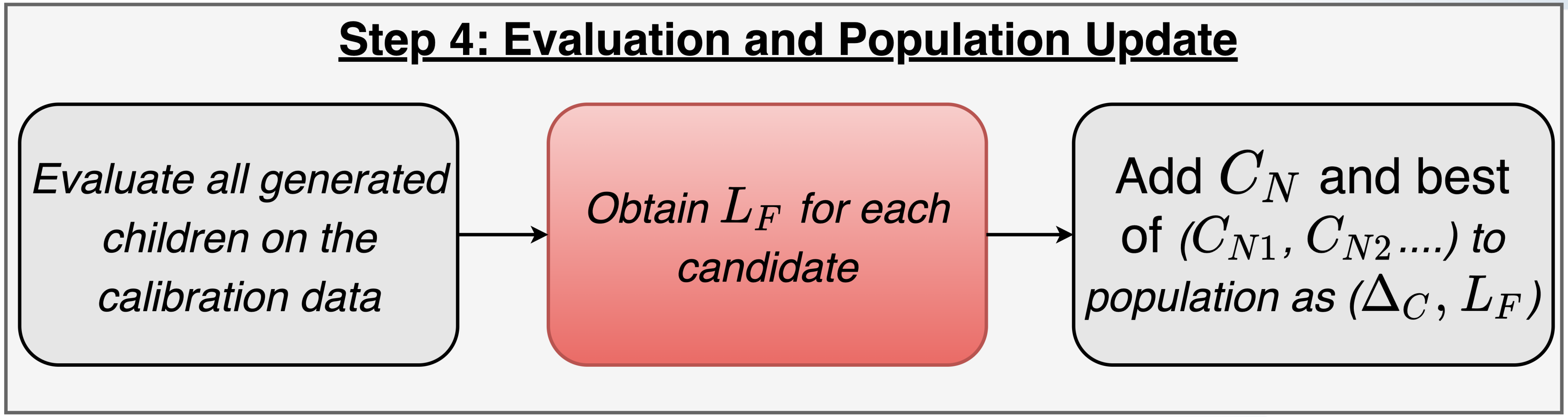


Step 3: Diversity Promoting Selection

We create additional random parents and use the regenerated child in the previous stage as the other parent to generate five diverse children.

Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

Automated Quantization Framework: Step 4



Activation Quantization

Activation quantization sensitivity closely aligns with that of the weight parameters producing them.

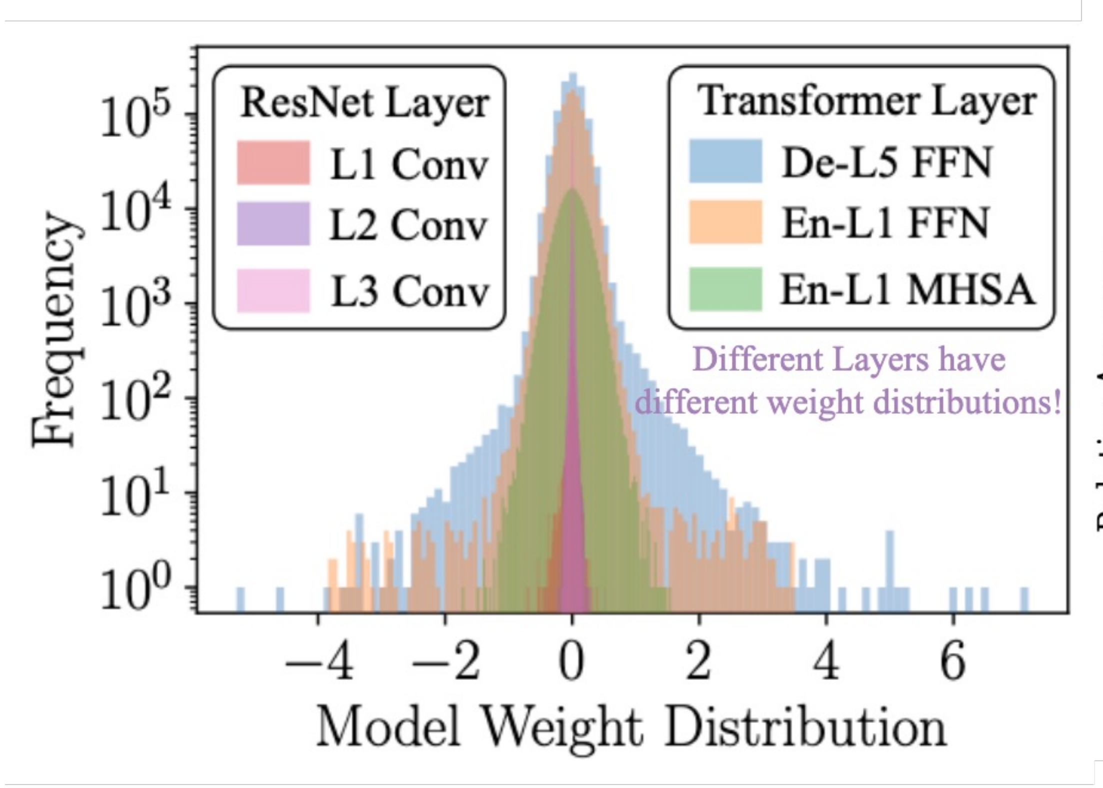
The output activation quantization parameters for layer i are determined as,

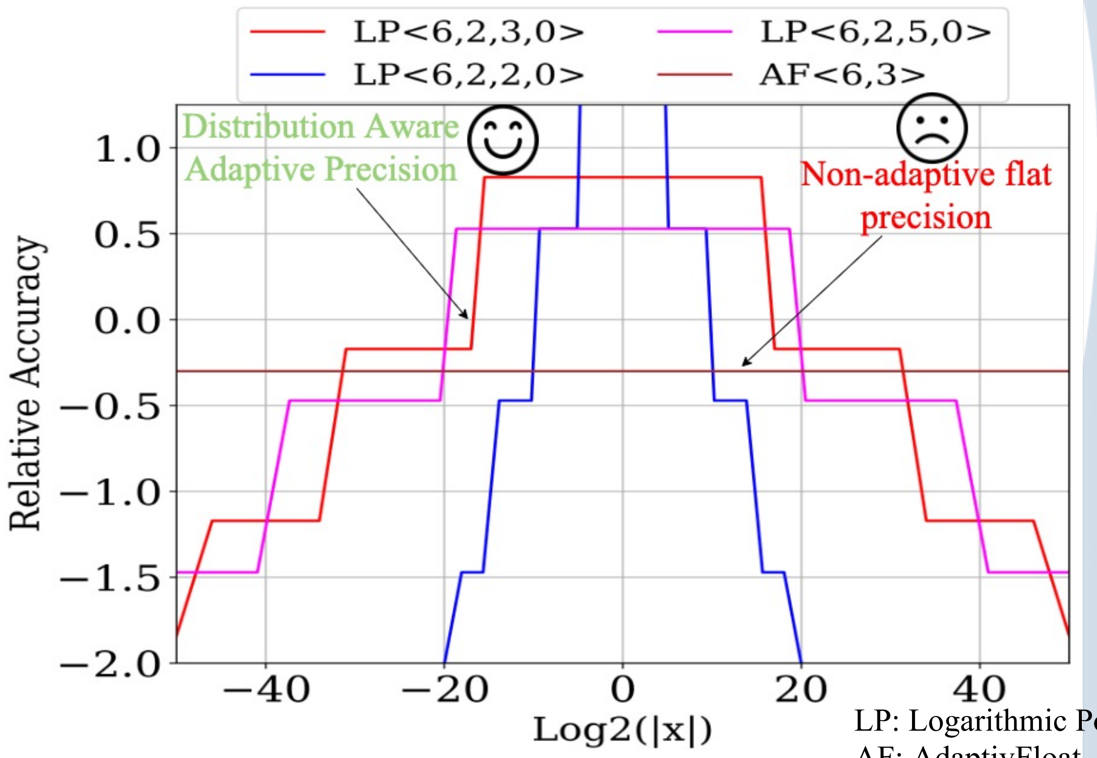
$$b_{act}[i] = \mathbf{min}(8, b[i] \times 2) \text{ and } \gamma_{act}[i] = \gamma_{act}[i-1] + \gamma[i]$$

Source: Ramachandran, A, et al. "Algorithm-Hardware Co-Design of Distribution-Aware Logarithmic-Posit Encodings for Efficient DNN Inference." arXiv preprint arXiv:2403.05465 (2024).

SRC

Challenges with Traditional Quantization



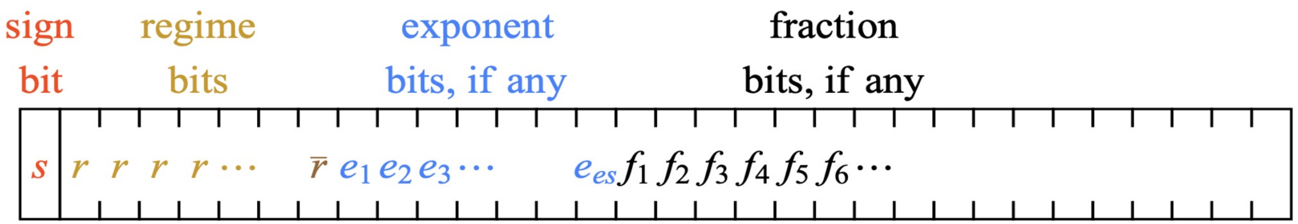


LP: Logarithmic Posit
AF: AdaptivFloat

- Uniform Quantization techniques lack the dynamic range and distributional variance required of DNNs.
- Floating-point based non-uniform quantization techniques fail to adapt to the different DNN distributions and have flat accuracy.
- Vector Quantization is adaptive but introduces additional codebook overhead.

SRC

Next Generation Arithmetic: Posits



Generic n-bits Posit format

float	Sign	Exponent (Size ES)		Fraction (Size F)
	s	$e_1e_2e_3 \dots e_{es}$		$f_1f_2f_3 \dots f_F$
posit	Sign	Regime (Run Length K)	Exponent (if any) (Size ES)	Fraction (if any)
	s	$r_1r_2r_3 \dots \overline{r_k}$	$e_1e_2e_3 \dots$	$f_1f_2f_3 \dots$

$$\texttt{posit val} = (-1)^{sign} * (2^{2^{es}})^k * 2^{expo} * 1.frac$$

- Possesses the unique field called regime that dynamically varies between [2, n-1] bits. It dynamically varies the exponent and fraction fields to give tapered accuracy and varied dynamic ranges.

- The regime is a run-length encoding of m 0s (1s) terminated by a 1(0), respectively, or by the final bit. The regime value k is determined as $k = -m$ if the first bit of the regime is 0, or $k = m-1$ otherwise.

Logarithmic Posits

$$\boxed{x\langle n, es, rs, sf\rangle = (-1)^{sign} \times 2^{2^{es}\times k - sf} \times 2^{\mathbf{ulfx}}}$$

LP value decoding

- A composite datatype that blends the adaptability of Posits with the hardware-efficiency of Logarithmic Number System (LNS).

- Parameterizations introduced for adaptability:
- **Bits (n):** Identify optimal precision for a DNN layer.
- **Exponent Size (es):** Controls dynamic range.
- **Regime Size (rs):** Controls distribution shape.
- **Scale Factor (sf):** Adjusts distribution position.

- Express standard fraction and exponent in the logarithmic domain as a unified fixed-point exponent of the power of two as 2^{ulfx}, where ulfx=e+f.

SRC

Floating-Point v. Posit v. Logarithmic Posit Decoding

Binary Number: 01001110

Floating-Point (E4M3), Bias = 7

0_1001_110

Sign: +

E-b: *1001 - 0111* = 9 - 7 = 2

M: *1.110* = 1.75

Value = +1 * 1.75 * 2^2

Posit (ES = 2)

0_10_01_110

Sign: +

Regime: *10* = 0

E: *01* = 1

M: 1.110 = 1.75

$$\textbf{Value} = +1 * (2^{(2^2)})^0 * 2^1 * 1.75$$

Logarithmic Posit (ES = 1, RS = 7, SF = 0

0_10_0_1110

Sign: +

Regime: *10* = 0

ulfx: *0.1110* = 0.875

$$\textbf{Value} = +1 * (2^{(2^2)})^0 * 2^0 * 2^{0.875}$$

Floating-Point v. Posit v. Logarithmic Posit Decoding

Binary Number: 01001110

Floating-Point (E4M3), Bias = 7

0_1001_110

Sign: +

E-b: *1001 - 0111* = 9 - 7 = 2

M: *1.110* = 1.75

Value = +1 * 1.75 * 2^2

Posit (ES = 2)

0_10_01_110

Sign: +

Regime: *10* = 0

E: *01* = 1

M: 1.110 = 1.75

$$\textbf{Value} = +1 * (2^{(2^2)})^0 * 2^1 * 1.75$$

Logarithmic Posit (ES = 1, RS = 7, SF = 0

0_10_0_1110

Sign: +

Regime: *10* = 0

ulfx: *0.1110* = 0.875

$$\textbf{Value} = +1 * (2^{(2^2)})^0 * 2^0 * 2^{0.875}$$

Floating-Point v. Posit v. Logarithmic Posit Decoding

Binary Number: 01001110

Floating-Point (E4M3), Bias = 7

0_1001_110

Sign: +

E-b: *1001 - 0111* = 9 - 7 = 2

M: *1.110* = 1.75

Value = +1 * 1.75 * 2^2

Posit (ES = 2)

0_10_01_110

Sign: +

Regime: *10* = 0

E: *01* = 1

M: 1.110 = 1.75

$$\textbf{Value} = +1 * (2^{(2^2)})^0 * 2^1 * 1.75$$

Logarithmic Posit (ES = 1, RS = 7, SF = 0

0_10_0_1110

Sign: +

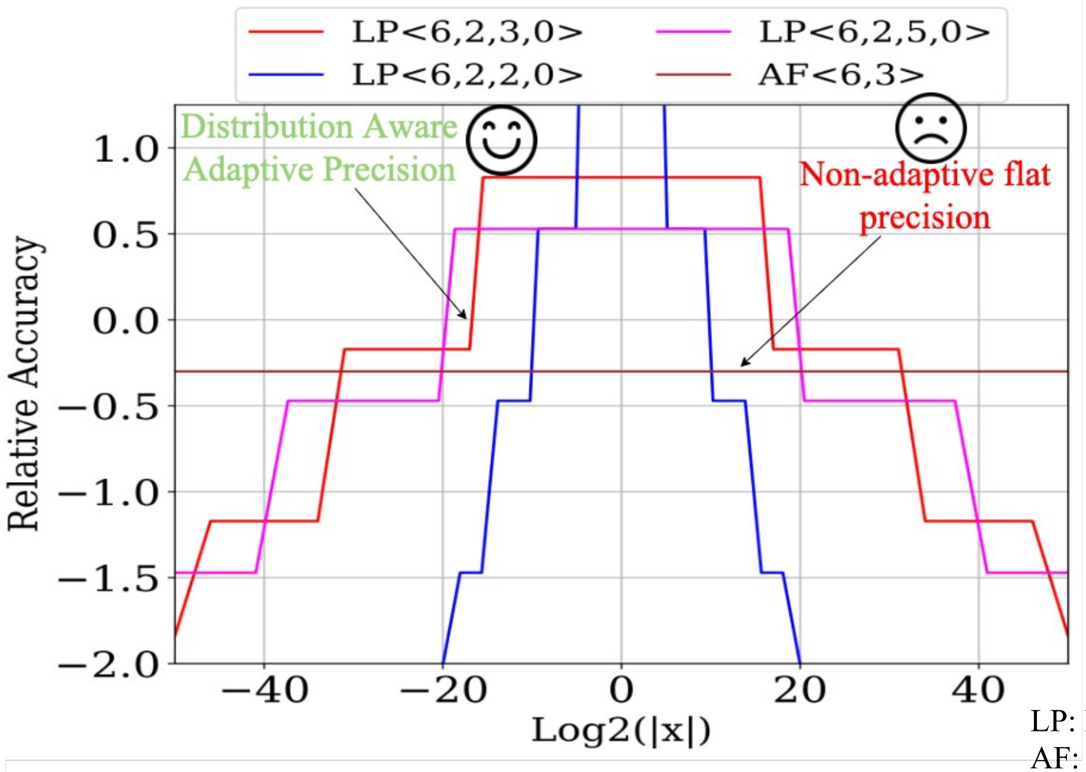
Regime: *10* = 0

ulfx: *0.1110* = 0.875

$$\textbf{Value} = +1 * (2^{(2^2)})^0 * 2^0 * 2^{0.875}$$

SRC

Advantages of Logarithmic Posits

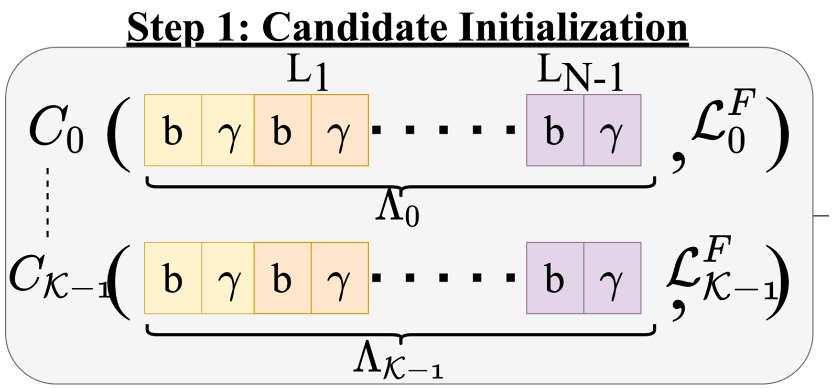


LP: Logarithmic Posit
AF: AdaptivFloat

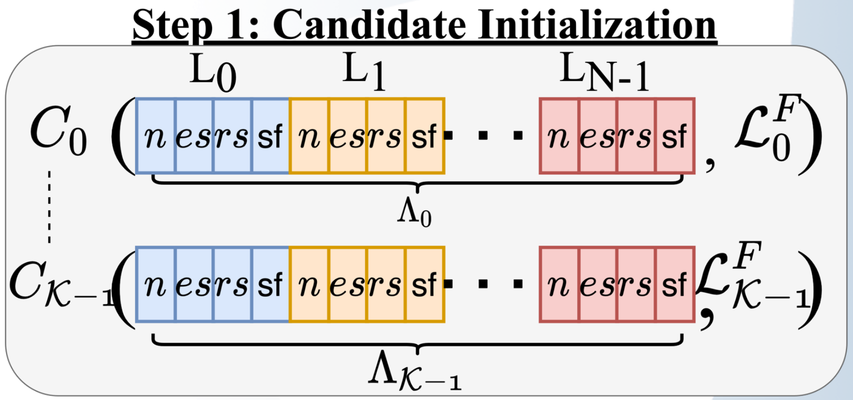
- By parameterizing the individual bitfields of LP, we are able to adapt the LP representation distribution to the required DNN data distribution.
- While also enjoying the hardware efficiency of LNS.

Changes in Algorithm to support LP

Uniform Quantization Candidates



Logarithmic Posit Quantization Candidates

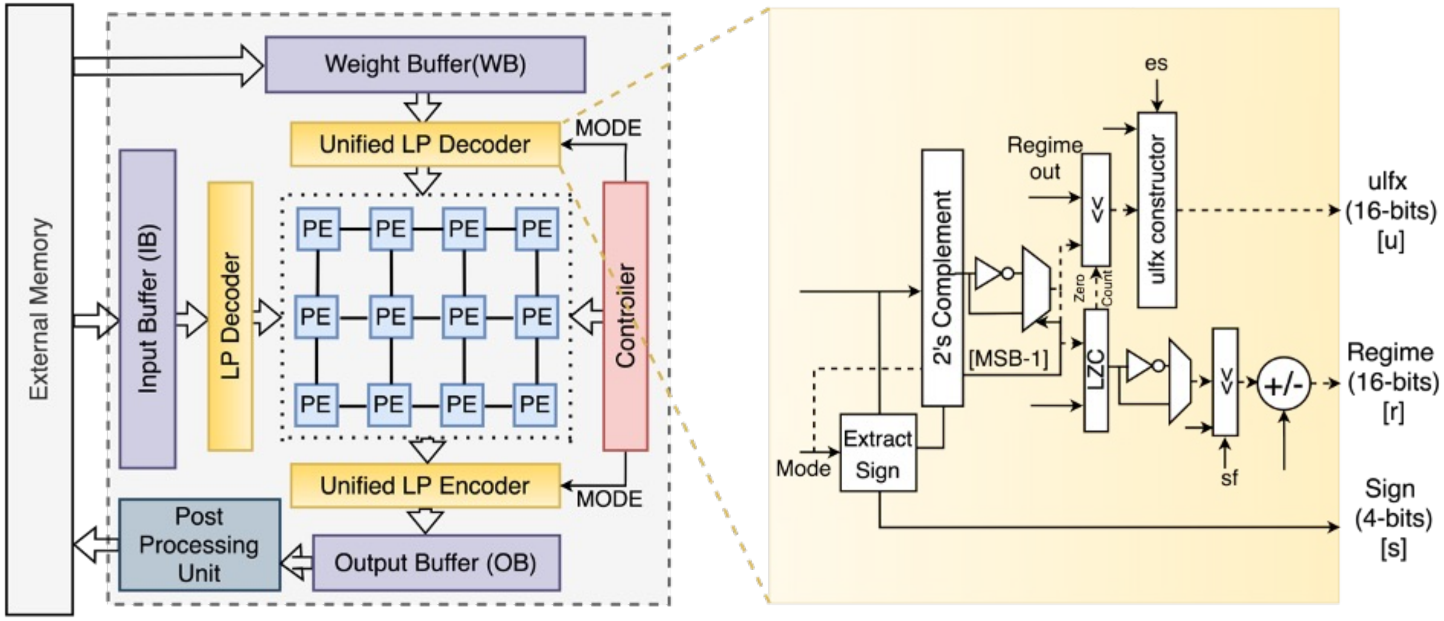


Our algorithm is general and can be easily applied to any quantization format by simply changing the candidate vector!

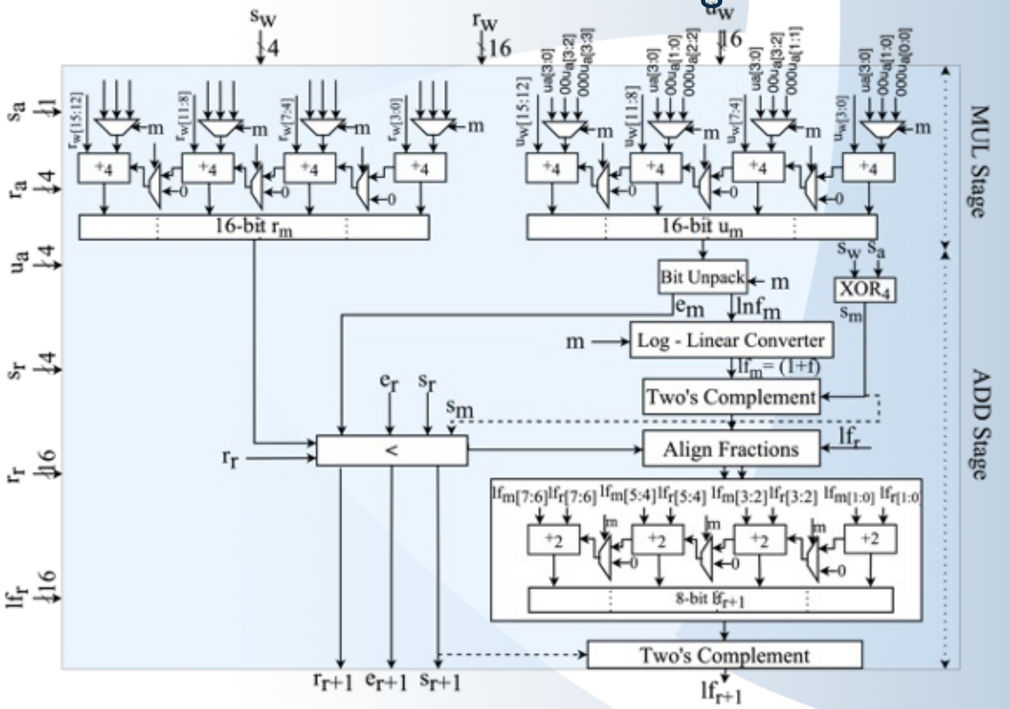
SRC

Hardware: Logarithmic Posit Accelerator

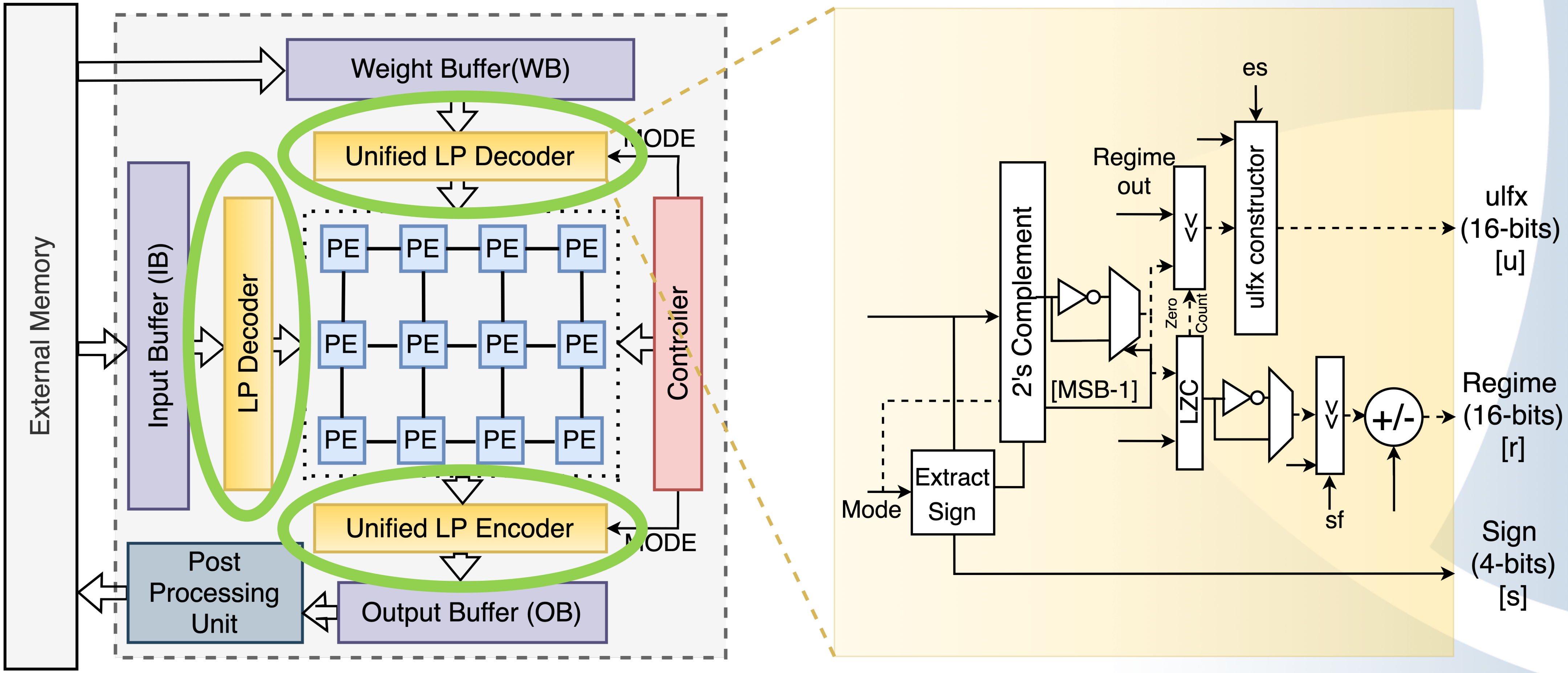
Modifications to a WS Systolic Array



Mixed-Precision PE Design



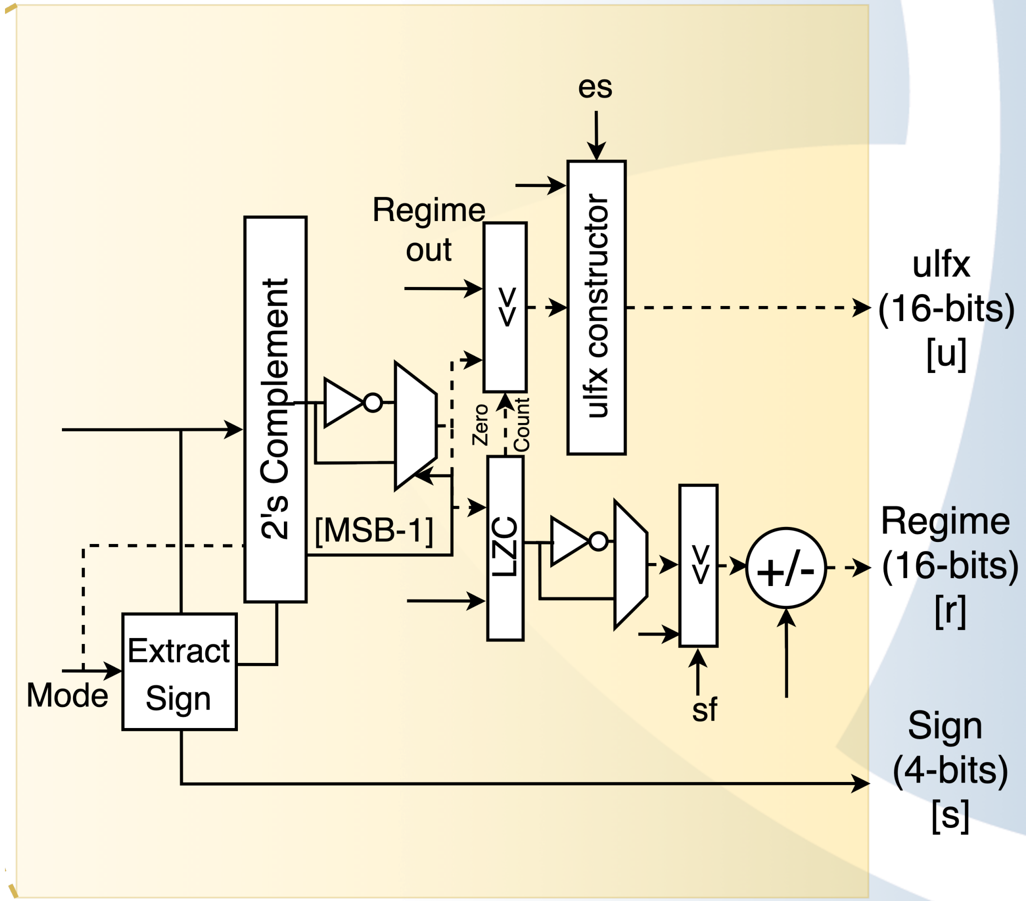
Modifications to Systolic Array



Modifications to Systolic Array

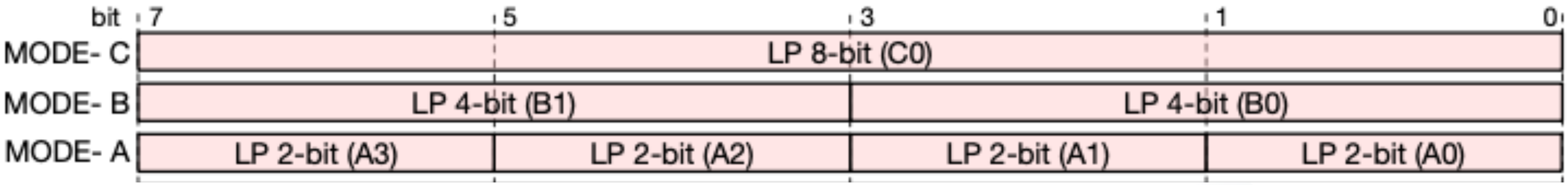
Decoder/Encoder Overhead is 1.03% of compute area for an 8x8 systolic array.

Exhibits weak scaling!!



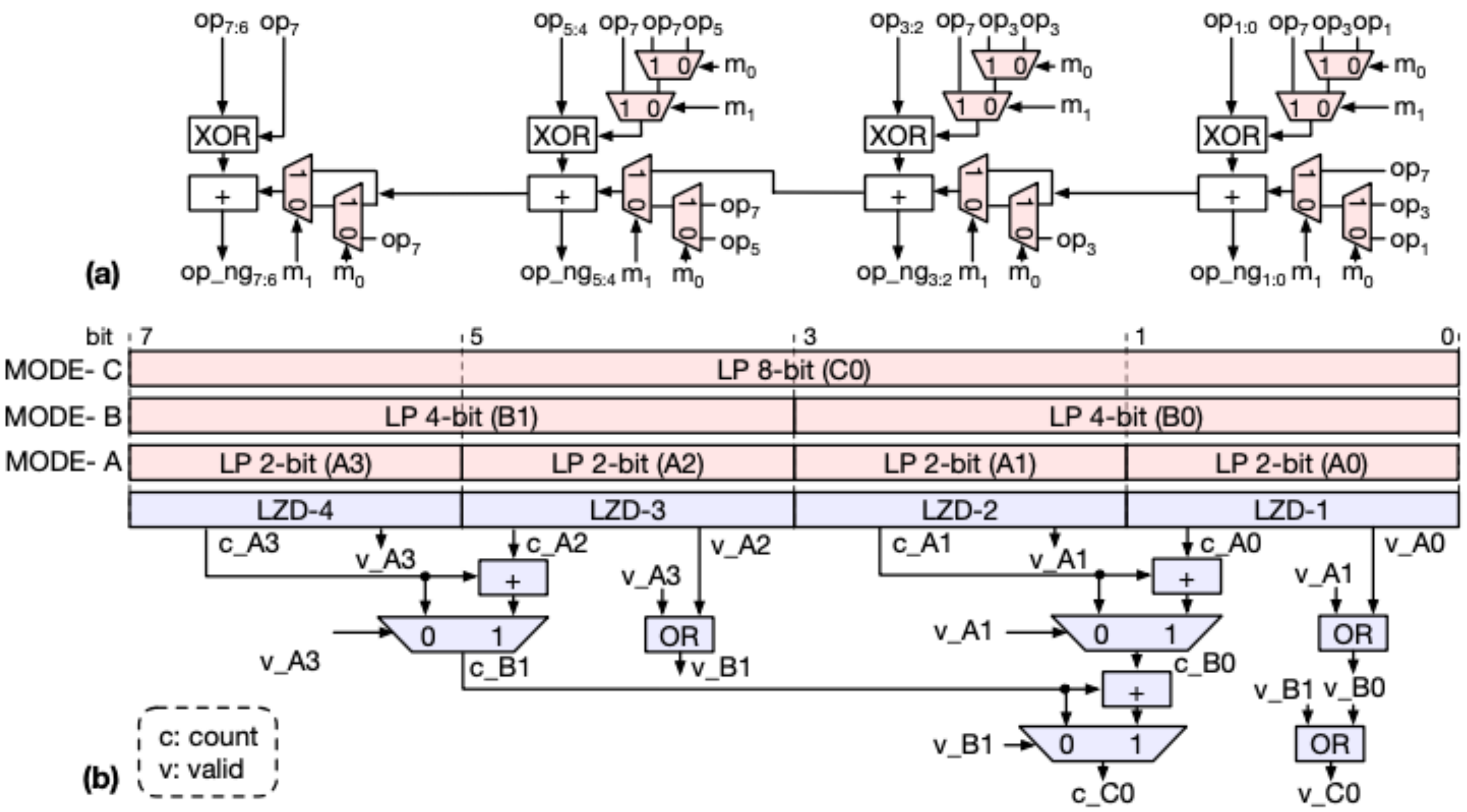
Concept of MODE

The MODE field is used to identify the precision for computation!



*Please refer the paper for more details on this and for mixed-precision decoder circuits

Implementing Mixed-Precision Components

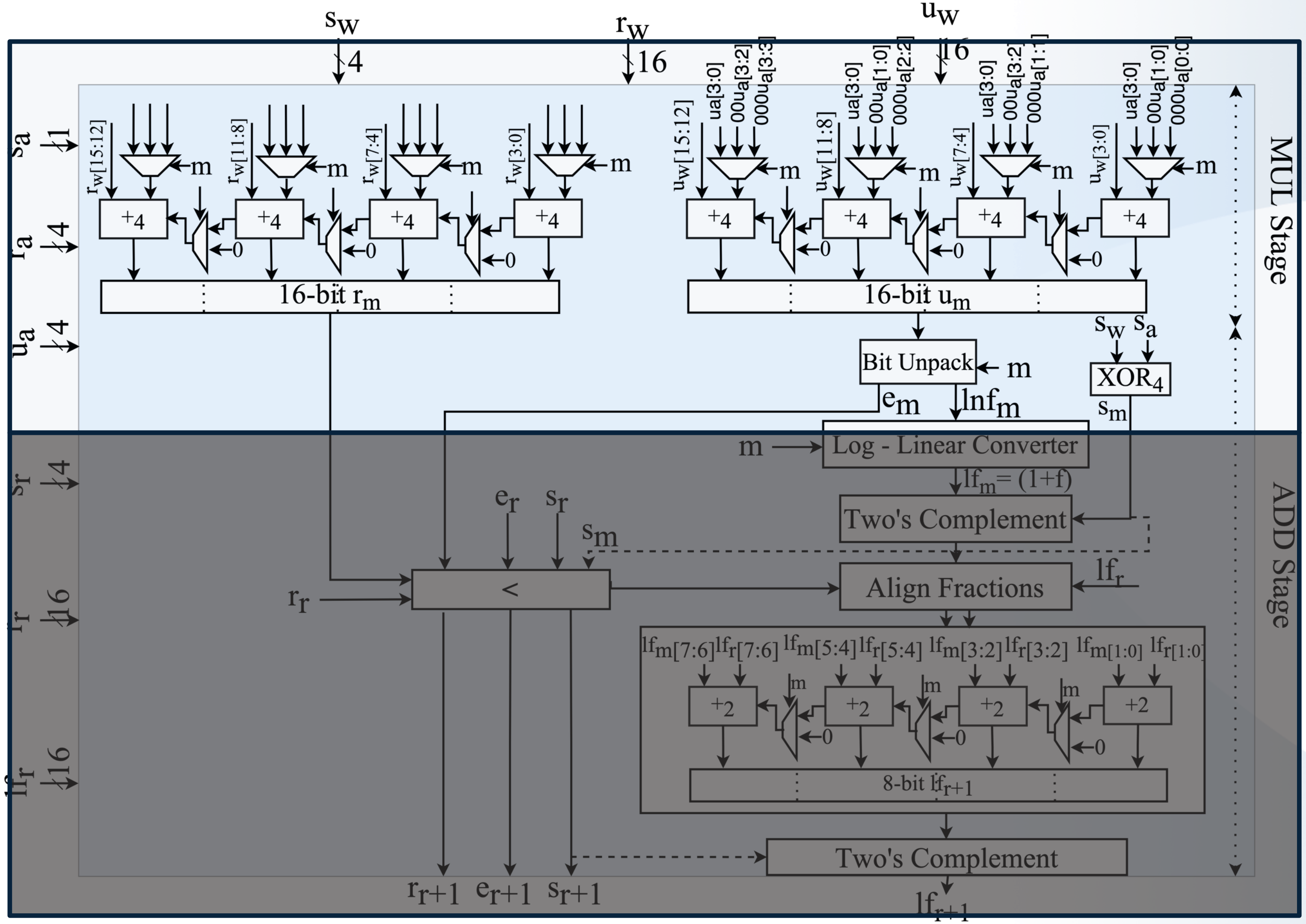


MP 2's Complementor

MP Leading Zero Counter (LZC)

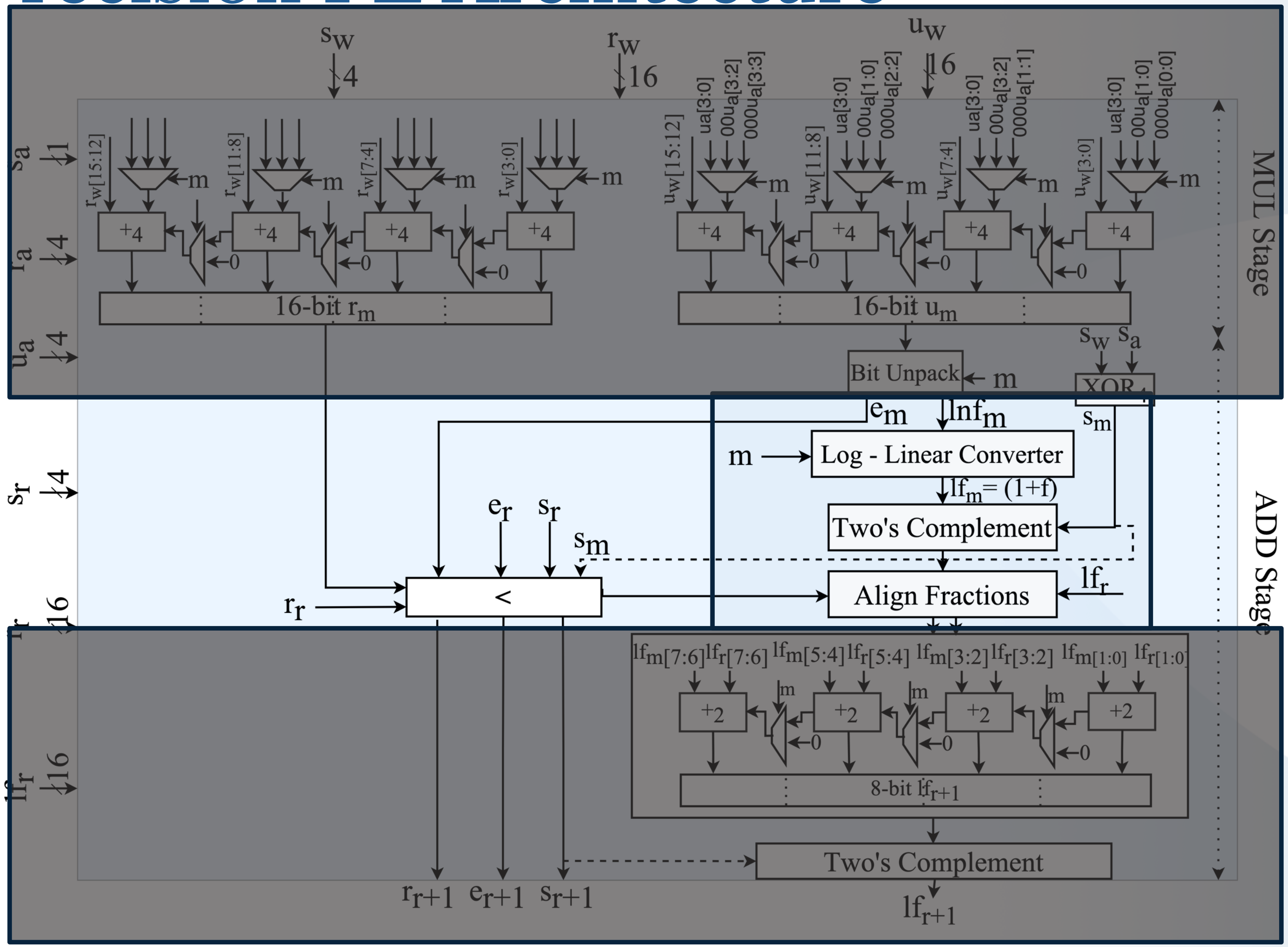
Mixed-Precision PE Architecture

Mixed-precision LP multiply-accumulate unit made entirely of 4-bit integer adder building blocks.

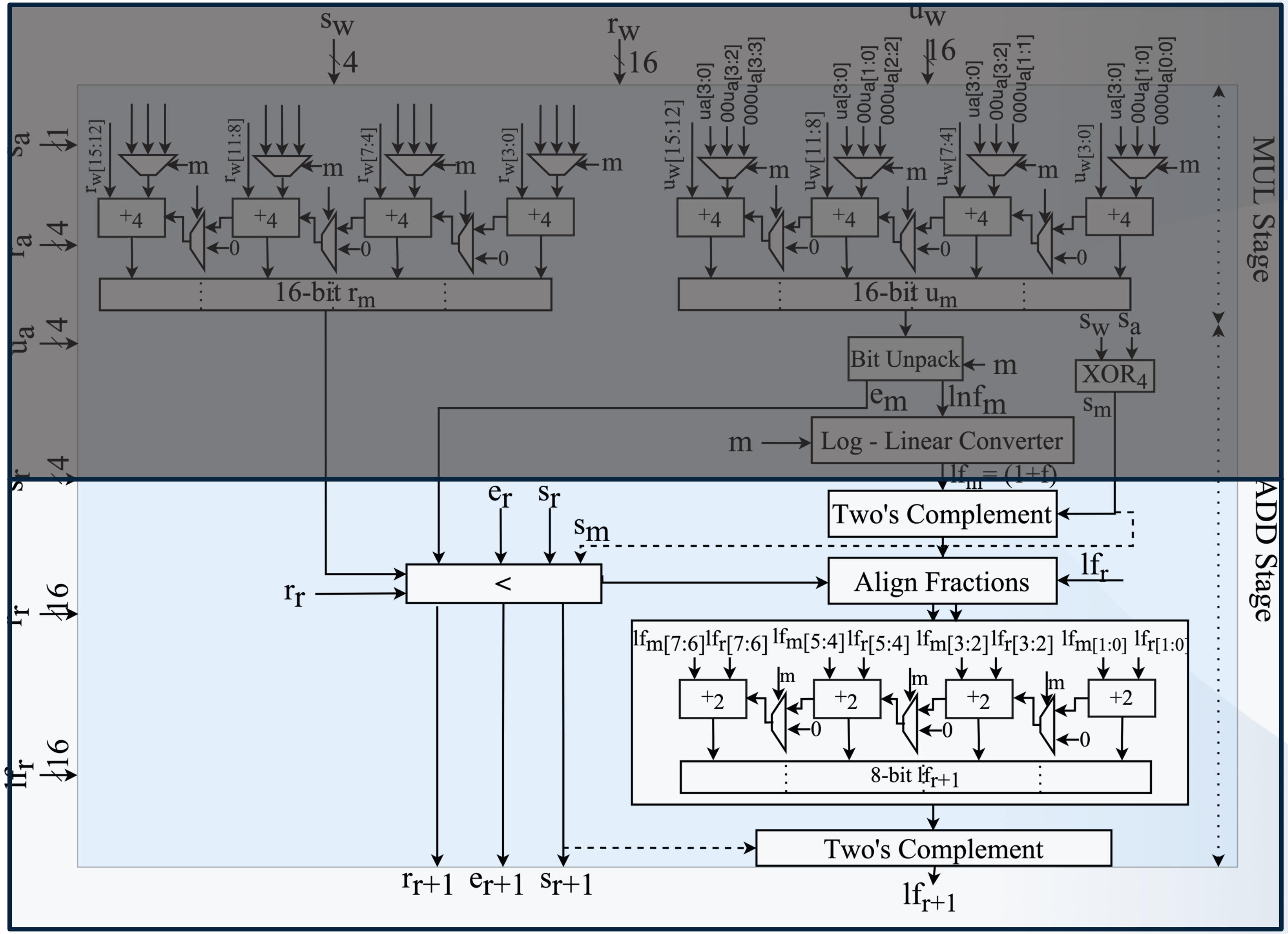


Mixed-Precision PE Architecture

Log-Linear Converter Implemented as Combinational Logic!

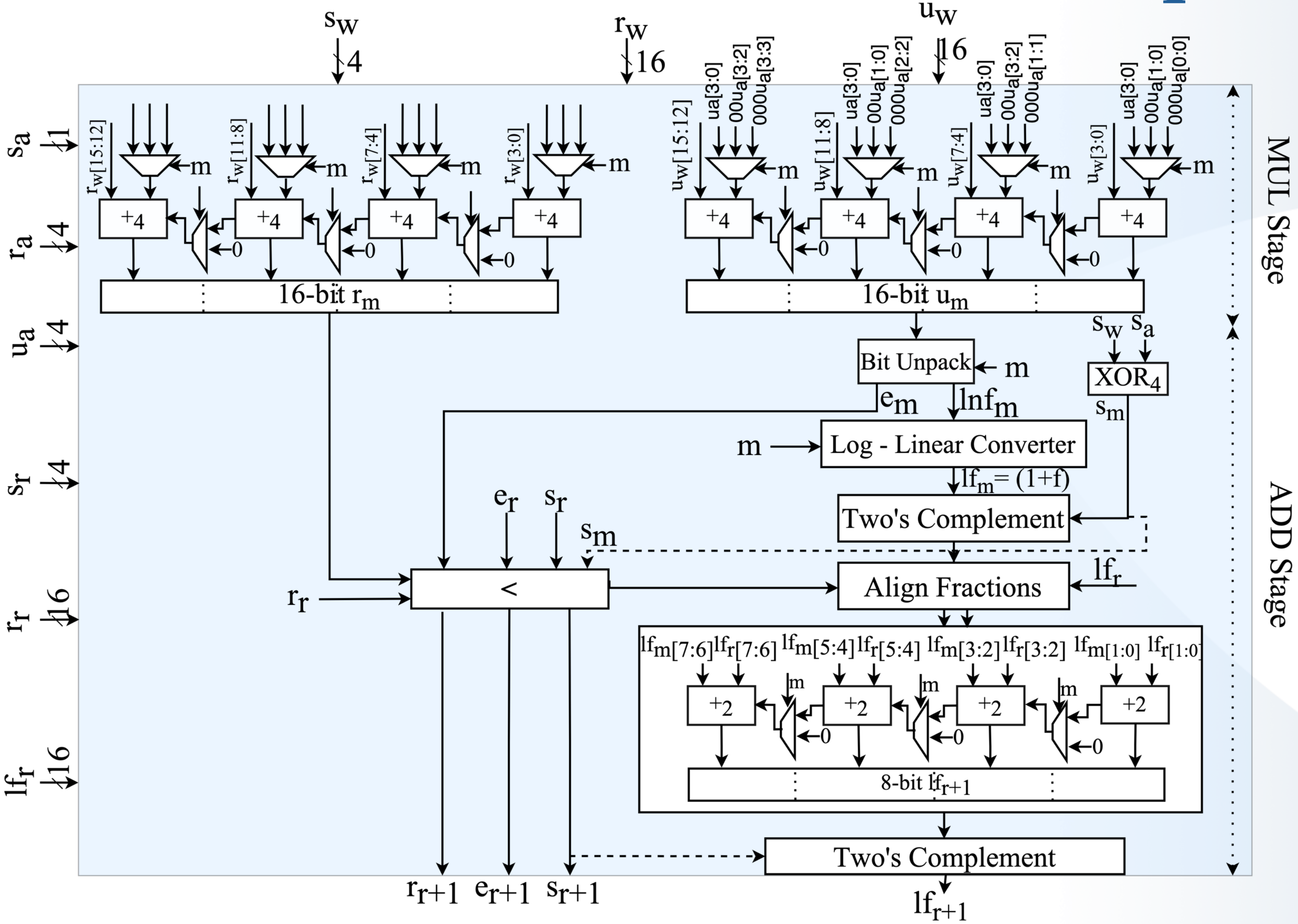


Mixed-Precision PE Architecture

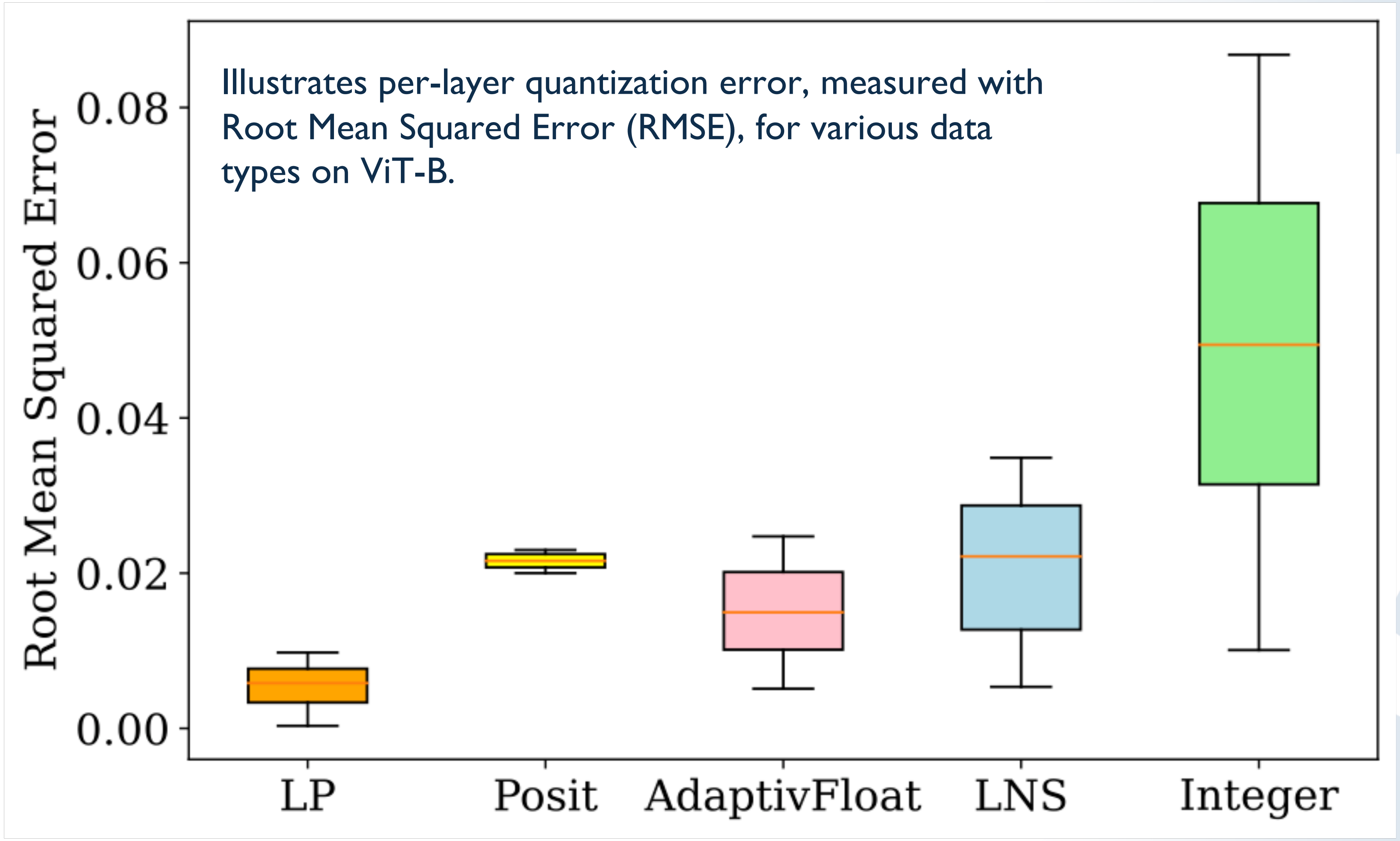


Linear Domain Addition made up of 2-bit adders.

Mixed-Precision PE Architecture: Complete Flow

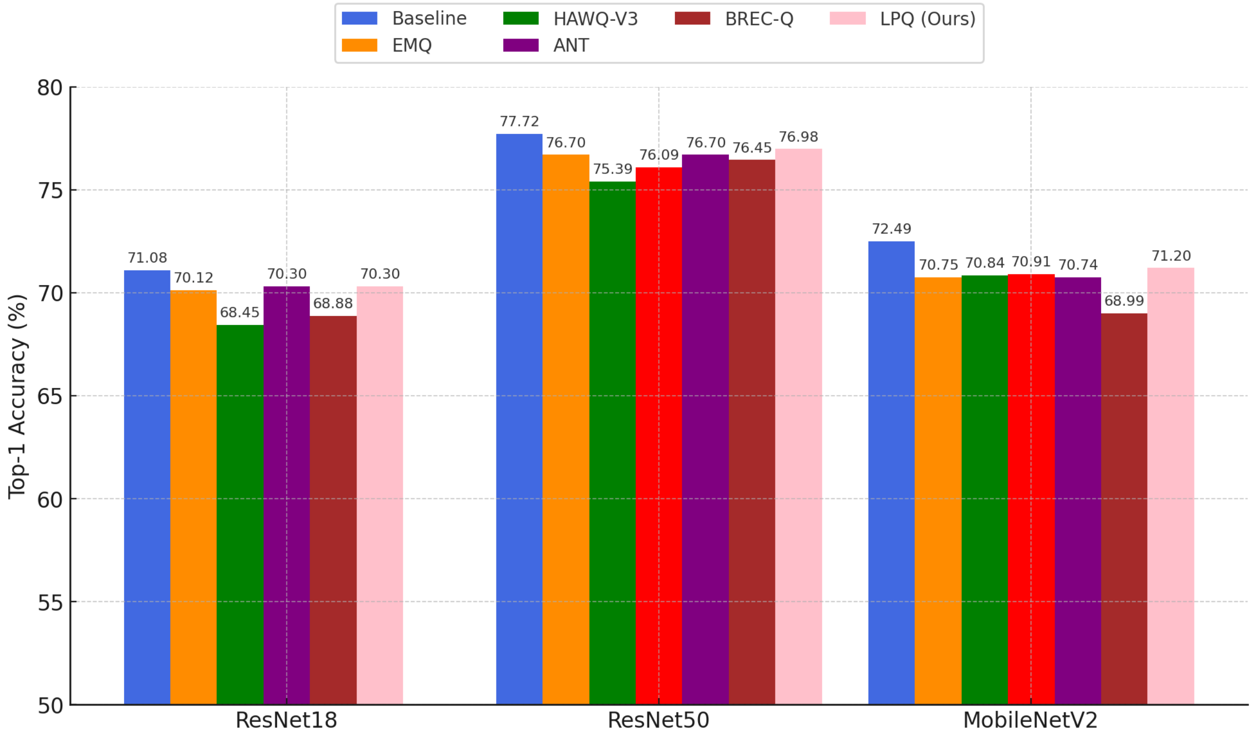


Number Format Comparison

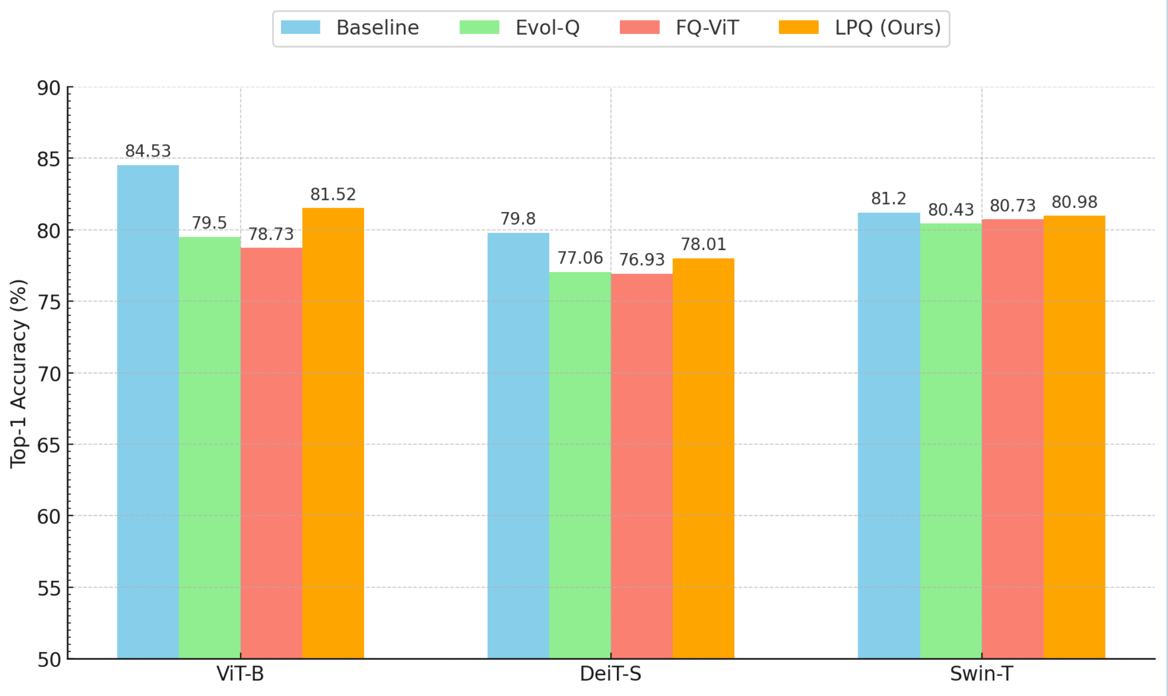


Mixed-Precision Quantization Results: CNN & ViT

CNNs

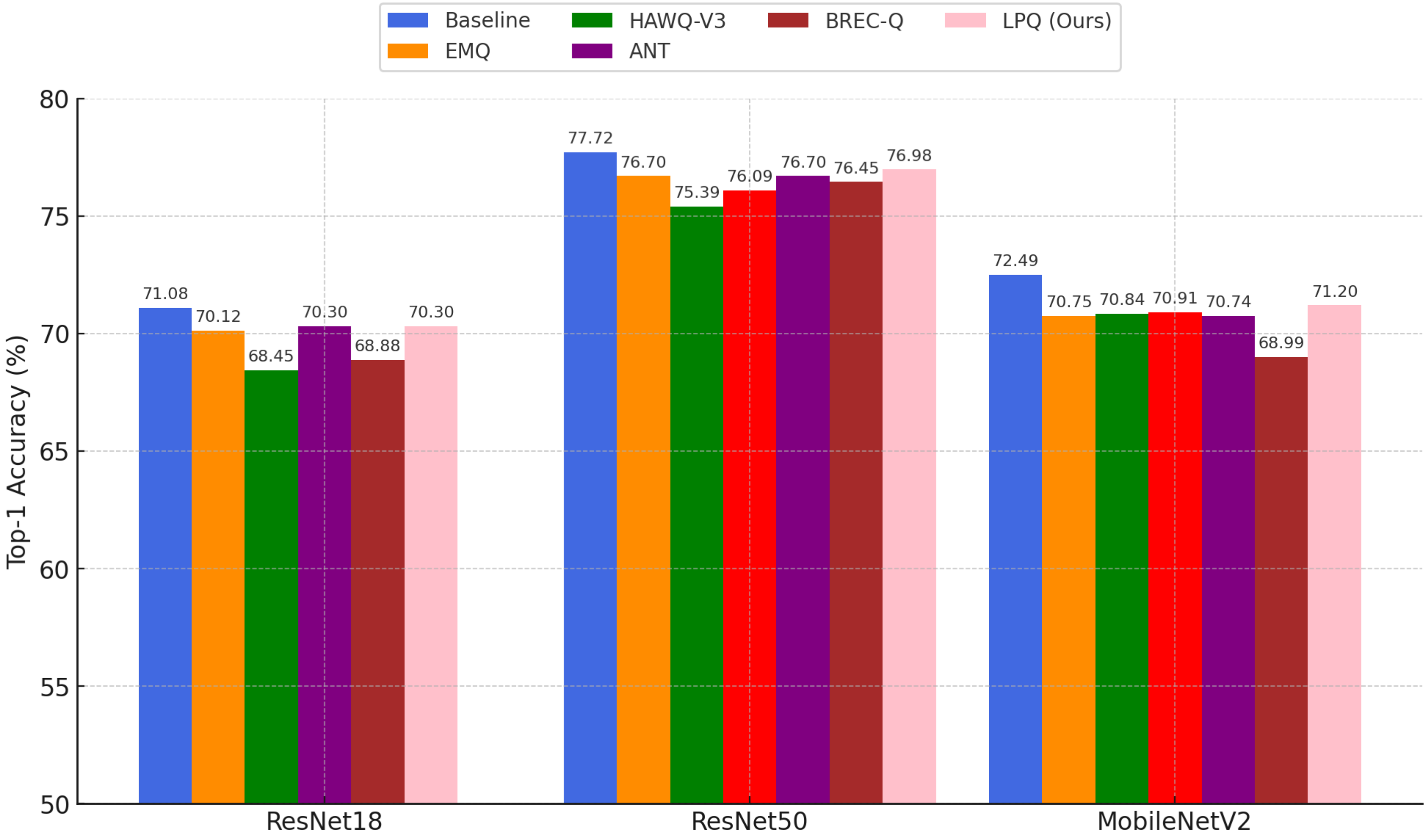


ViTs



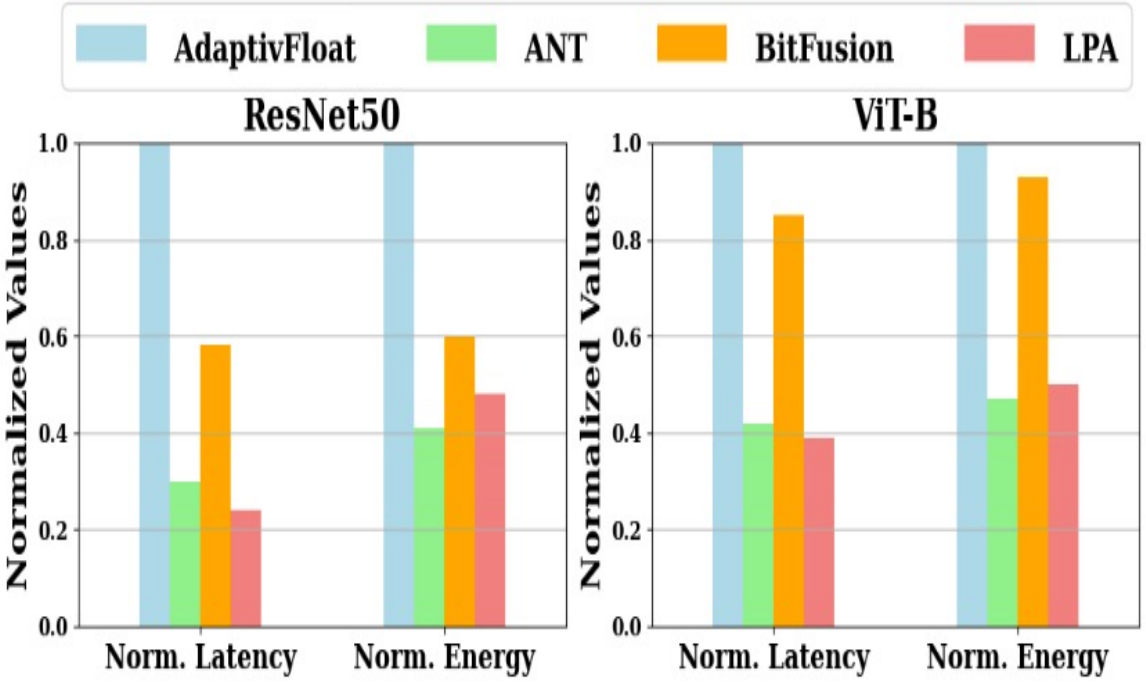
SRC

Mixed-Precision Quantization Results: CNN & ViT



- **Highlights:**
- On average <1% accuracy degradation compared to FP baseline.
- Mixed-Precision LPQ achieves average weight/activation bitwidth of 4.2/5.5.
- 90% reduction in model size.

Logarithmic Posit Accelerator Performance



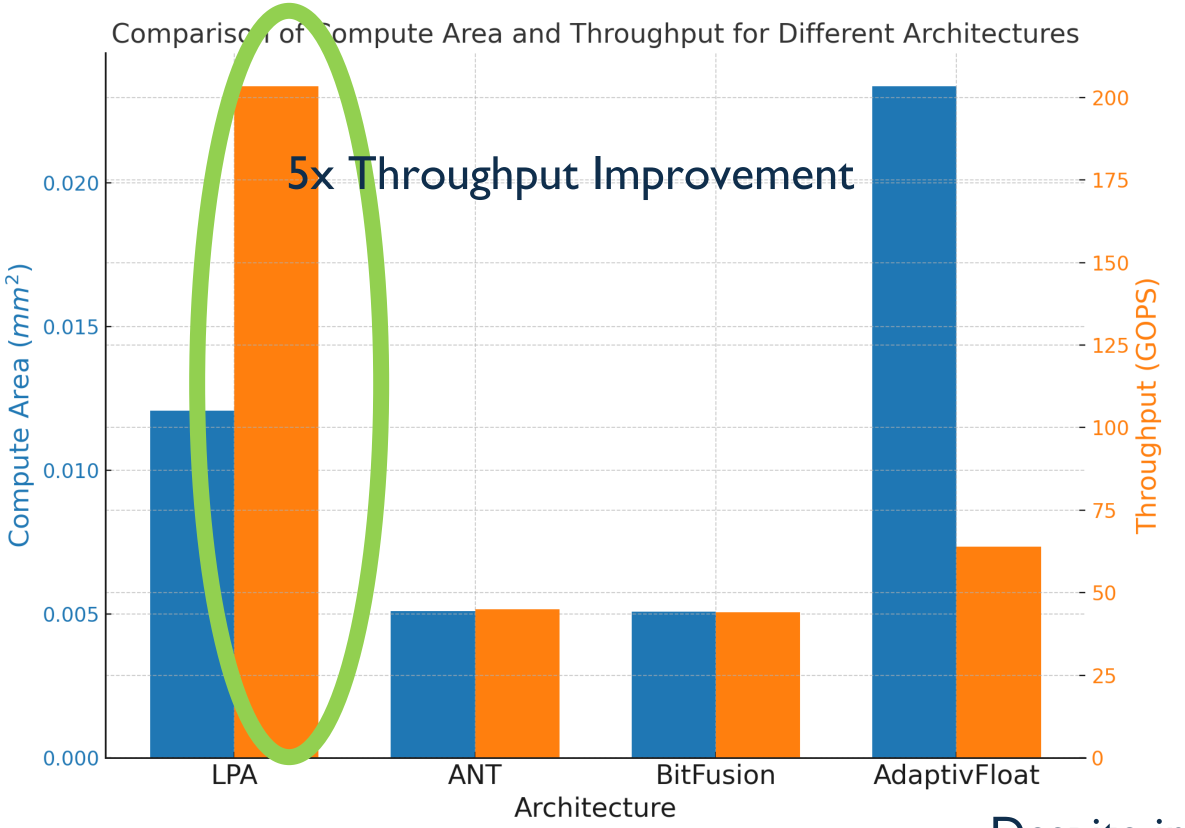
LPA exhibits the lowest latency across models, with a modest increase in energy consumption over ANT attributed to overheads due to native mixed- precision support and conversion logic.

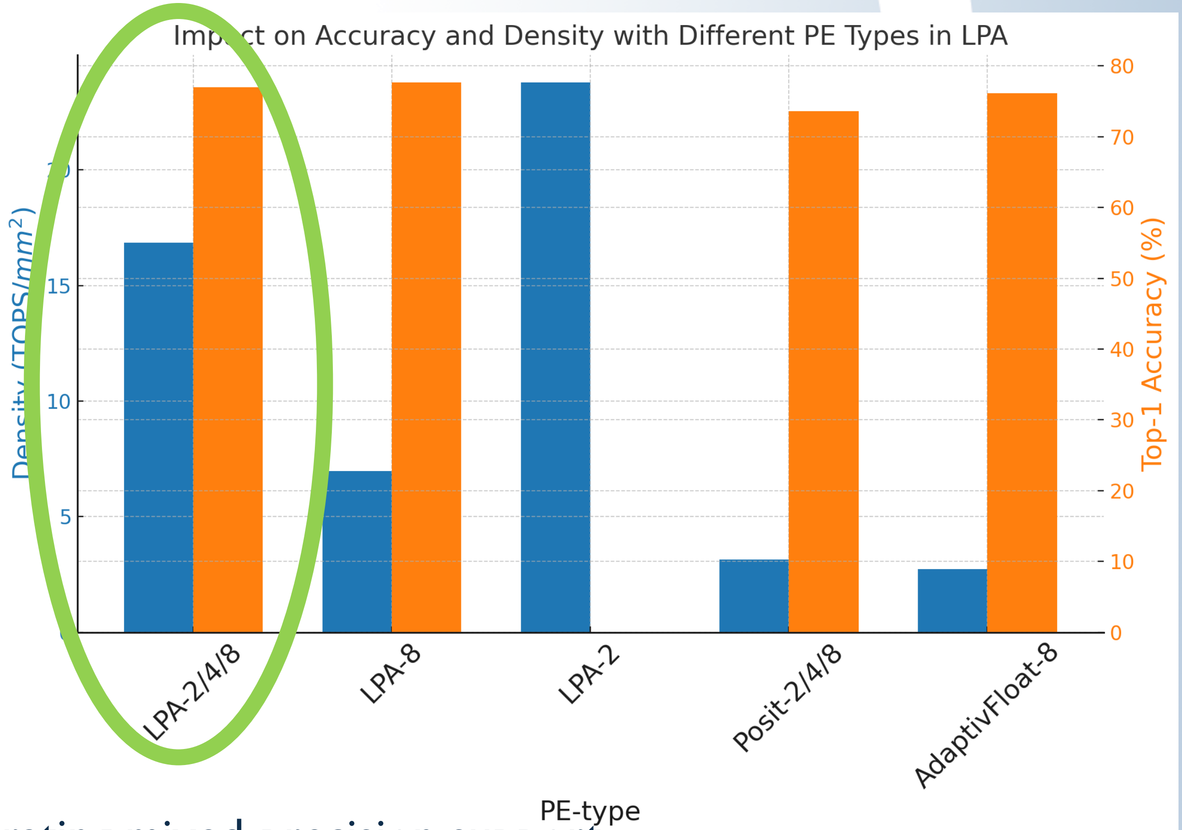
Architecture	Component (Area)	Compute Area (μm^2)	Throughput (GOPS)	Compute Density (TOPS/mm^2)	Total Area (mm^2)
LPA	Decoder(5.2 μm^2)	12078.72	**203.4**	**16.84**	4.212
	Encoder (9.4 μm^2)				
	2/4/8-bit PE (187.43 μm^2)				
ANT	Decoder(4.9 μm^2)	5102.28	44.95	8.81	**4.205**
	4/8-bit Int PE (79.57 μm^2)				
BitFusion	2/4/8-bit PE	**5093.75**	44.01	8.64	**4.205**
AdaptivFloat	8-bit PE	23357.14	63.99	2.74	4.223

Despite ANT and BitFusion exhibiting lower area when compared with LPA for the same number of PEs, LPA results in proportionately higher performance per unit area (TOPS/$mm2$) for mixed-precision DNN inference.

SRC

Logarithmic Posit Accelerator Performance





Despite incorporating mixed-precision support, LPA-2/4/8 achieves accuracy tending to the ideal scenario for both metrics, demonstrating a balanced trade-off.

Summary

CONTEXT

Current Approach and Challenges

- Approach: Algorithm-Hardware Co-Design is a promising area towards efficient DNN inference.
- Challenges: Inefficient data formats and lack of generalizable automated techniques.

APPROACH

The proposed approach:

- Develop an adaptive, hardware-friendly data format.
- Identify existing limitations of automated quantization algorithms.
- Optimize existing hardware to support next-generation data formats.

IMPACT

How do current results advance SOTA?

- <1% Accuracy drop across DNN model families post-quantization with > 15% higher compression ratio than competing methods.
- 2x improvement in PPA and energy-efficiency.

Next Steps

- Extend: Verify adaptability and generalizability to LLMs and VLMs.
- Profile: Deepen understanding of data distribution of modern DNN models to improve existing data-format parameterization.
- Improve runtime of existing algorithm for faster quantization of large-scale models.

ACADEMIA • INDUSTRY • GOVERNMENT

CoCoSys

CENTER FOR THE
CO-DESIGN OF COGNITIVE SYSTEMS