HONGIK UNIVERSITY

2025/01/20

PopBin: Popcount Binarization for Lightweight Binary Neural Networks

Hyungdong Park, Inguk Yeo
Department of Computer Engineering

Contents

Background

Binary Neural Networks (BNNs)

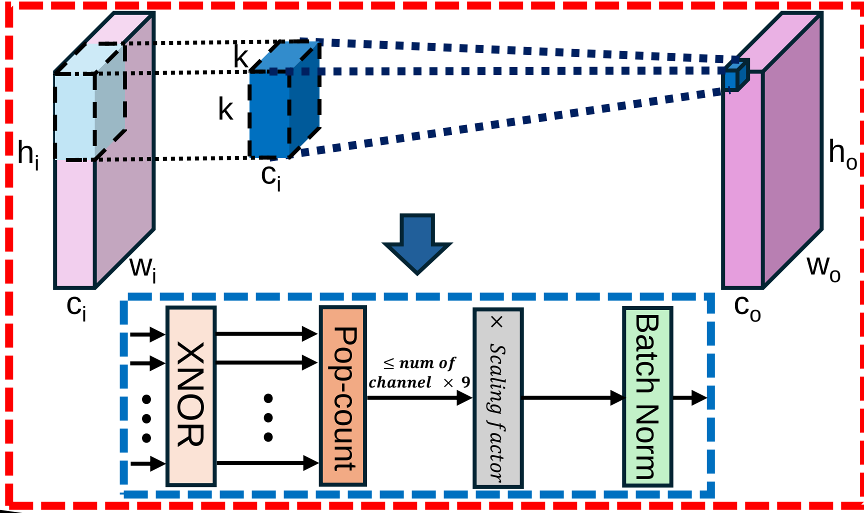
- XNOR-Net
  - XNOR operation-based binary neural network model. Description of its features and working principles.
- Bi-Real Net
  - The binarization technique used in Bi-Real Net and its performance improvements.
- ReActNet
  - The role of binarization and activation functions in ReActNet, along with related optimization techniques.

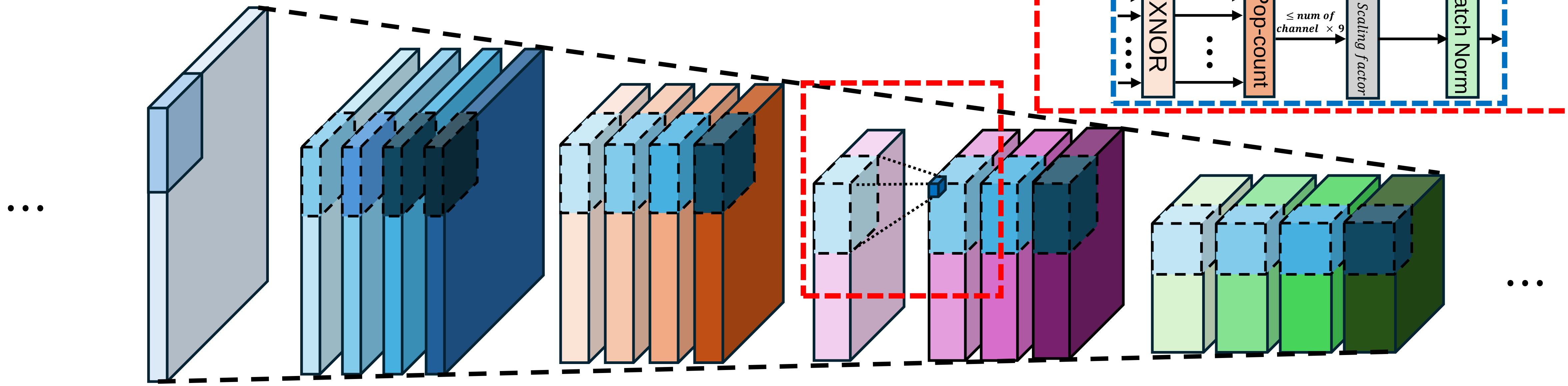
Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT)

- Post-Training Quantization (PTQ)
  - The basic concept of PTQ and its role in the binarization process of BNNs.
- Quantization-Aware Training (QAT)
  - Explanation of QAT techniques, differences from PTQ, and the advantages QAT provides for BNNs.

Background

- ReActNet-18 with CIFAR-10 using Xnor & Popcount





Num of channels	64	64	128	256	512	Pooling & FC
Num of layers	1	4	4	4	4	
Image size	64 × 32 × 32	64 × 32 × 32	128 × 16 × 16	256 × 8 × 8	512 × 4 × 4	
Operations	⊛	XNOR & Pop-Count & Multiplication & Batch Norm				
Activations and weights	$\mathbb{R}$	$\mathbb{R}$ (After BN) & Binarized values (1 or -1)				$\mathbb{R}$
# units of Mul		$c_o \times h_o \times w_o$				

Challenges Induced by Popcount Results in BNNs

Analysis of Latency Issues

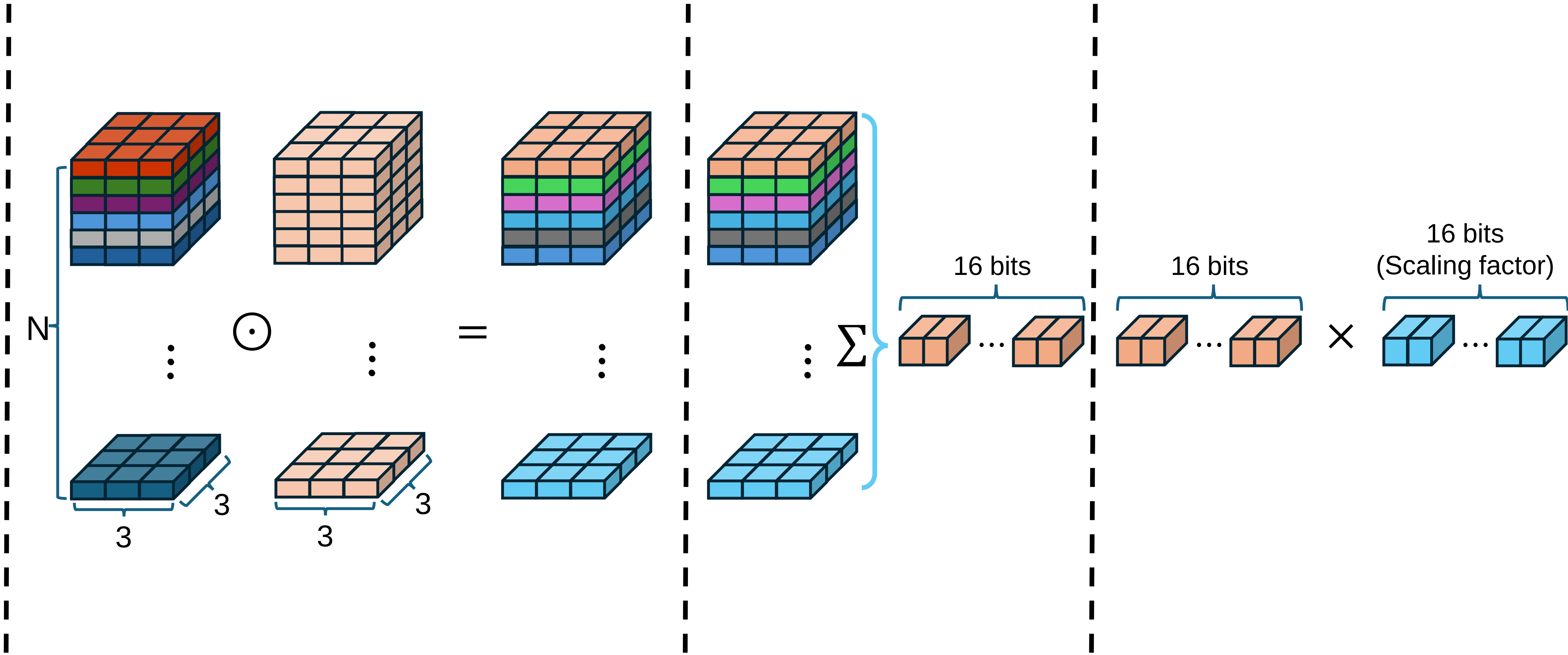
- Analyzing the latency impact that Popcount results have on BNNs' performance.

Popcount Latency Optimization

- Exploring hardware optimization techniques to reduce latency caused by Popcount results.

Challenges Induced by Popcount Results in BNNs

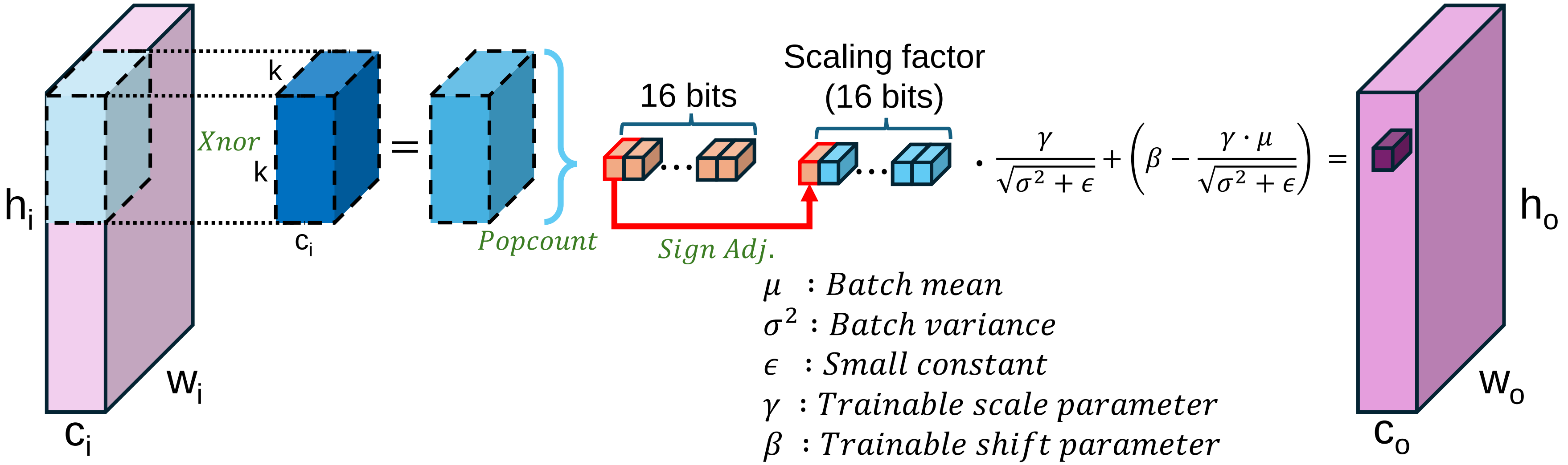
- Analysis of latency issues



	Xnor (Step 1)	*Popcount (Step 2)*	*Integer Mul (Step 3)*
Operations	*Xnor* ($\odot$)	*Popcount*	*Integer multiplication & Bit shift*
# operations	$N \times 3^2$	$(N \times 3^2) - 1$	1 *Integer multiplication* & 1 *Bit shift*

Challenges Induced by Popcount Results in BNNs

- Popcount Latency Optimization



Operations in QAT-popcount binarization ReActNet-18

Models	Operations	# Operations
ReActNet-18	Xnor & Popcount & Integer Multiplication & Bit shift	$channel \times kernel^2$ *Xnor* & $(channel \times kernel^2 - 1)$ *Popcount* & $c_o \times w_o \times h_o$ *Integer Multiplications* & *Bit shifts*
QAT-Popcount binarization ReActNet-18	Xnor & Popcount	None

Popcount Binarization Strategies

PTQ-Popcount Binarization

- A PTQ-based binarization method aimed at minimizing the impact of Popcount results. It is easy to apply but leads to a significant drop in accuracy.

Simple QAT-Popcount Binarization

- A method that uses QAT to improve the Popcount issue. It improves accuracy, but still falls short of the original model's performance.

QAT-Popcount Binarization

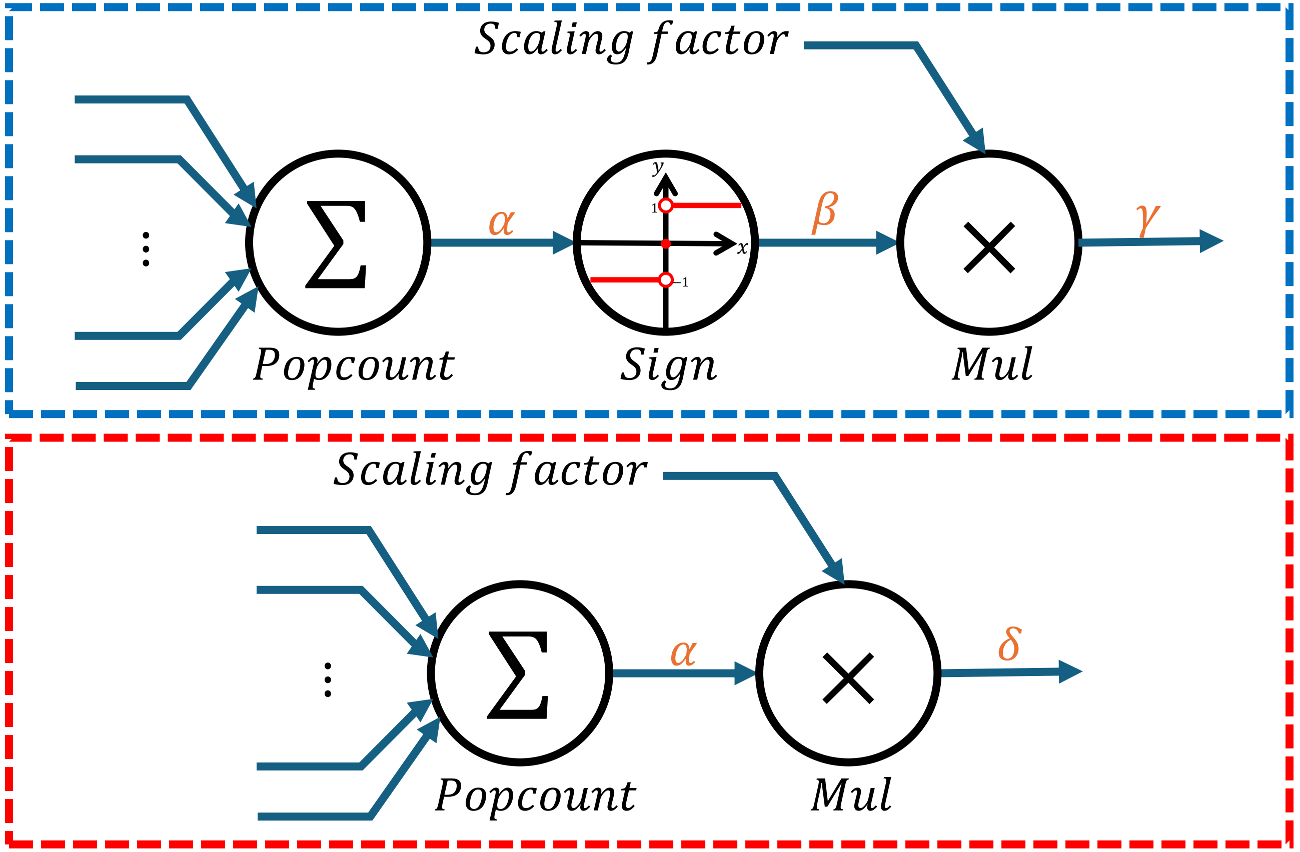
- An advanced binarization technique using QAT. It maintains accuracy within 1% of the original model while optimizing performance.

Latency Reduction through Popcount Optimization

- An optimization strategy to address latency issues caused by Popcount operations, enhancing overall system efficiency.

Popcount Binarization Strategies

- PTQ-Popcount Binarization



$|\alpha| \leq (channel\_num \times kernel^2)$

$\beta = \pm 1$

$\gamma = \pm\, scaling\ factor$

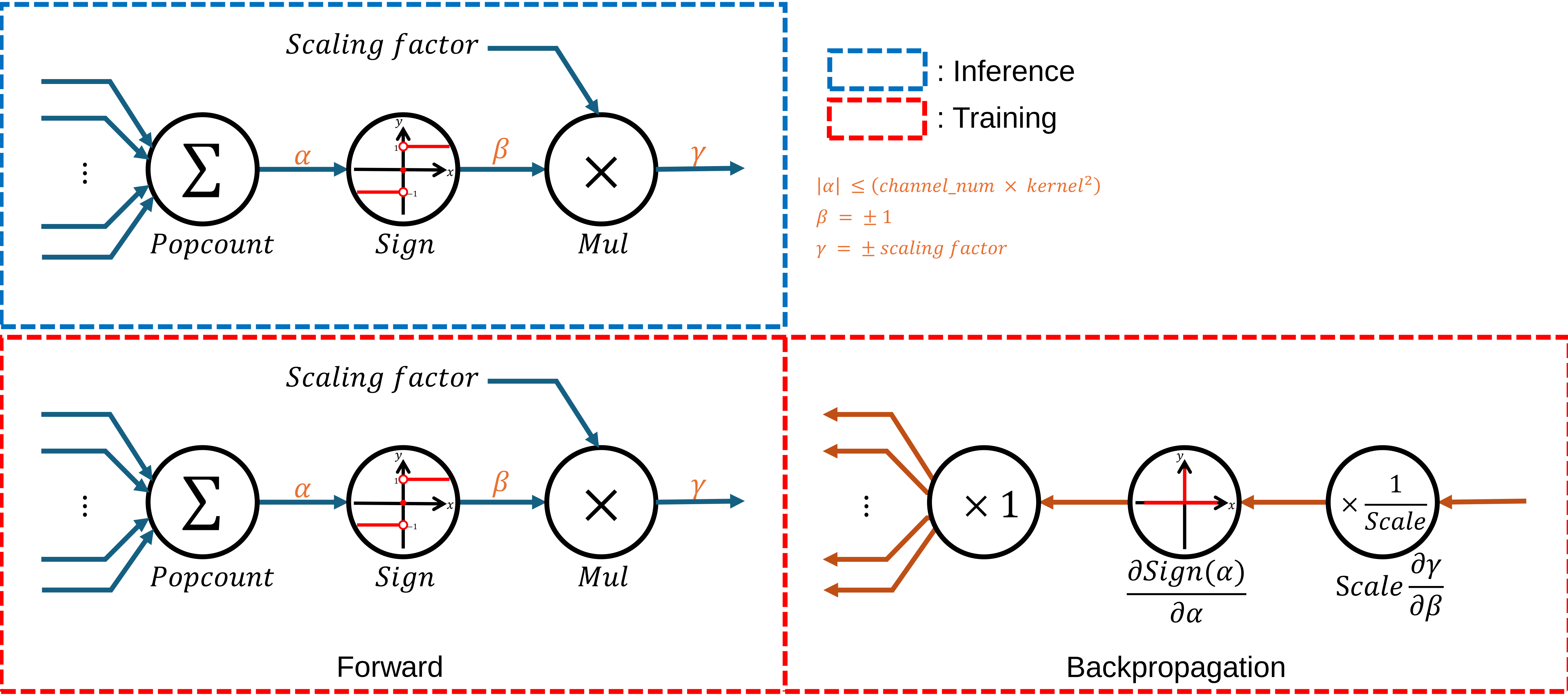
$\delta = \pm(scaling\ factor \times channel\_num \times kernel^2)$

: Inference

: Training

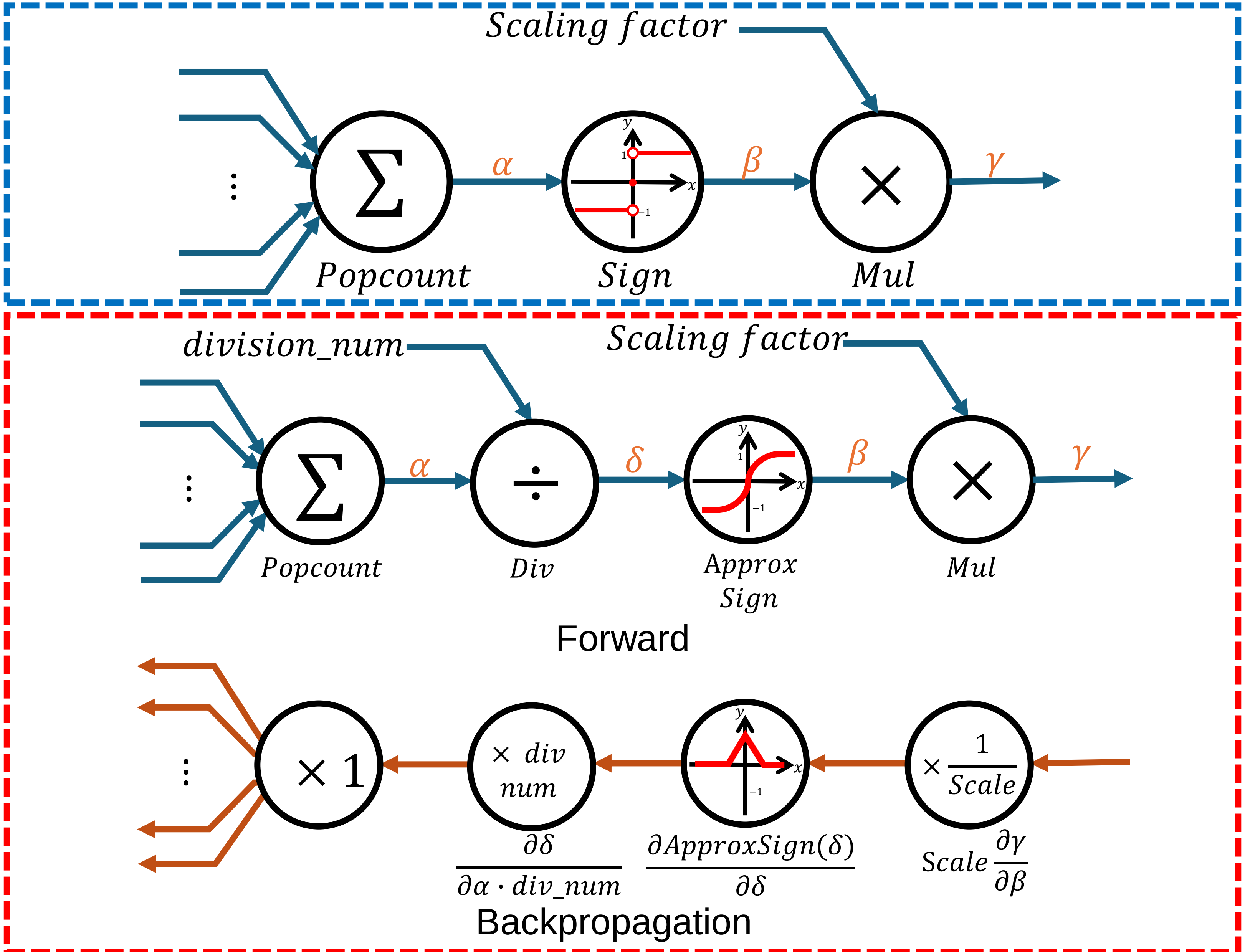
Popcount Binarization Strategies

- Simple QAT-Popcount Binarization



Popcount Binarization Strategies

- QAT-Popcount Binarization



$|\alpha| \leq (channel\_num \times kernel^2)$

$\beta = \pm 1$

$\gamma = \pm\, scaling\ factor$

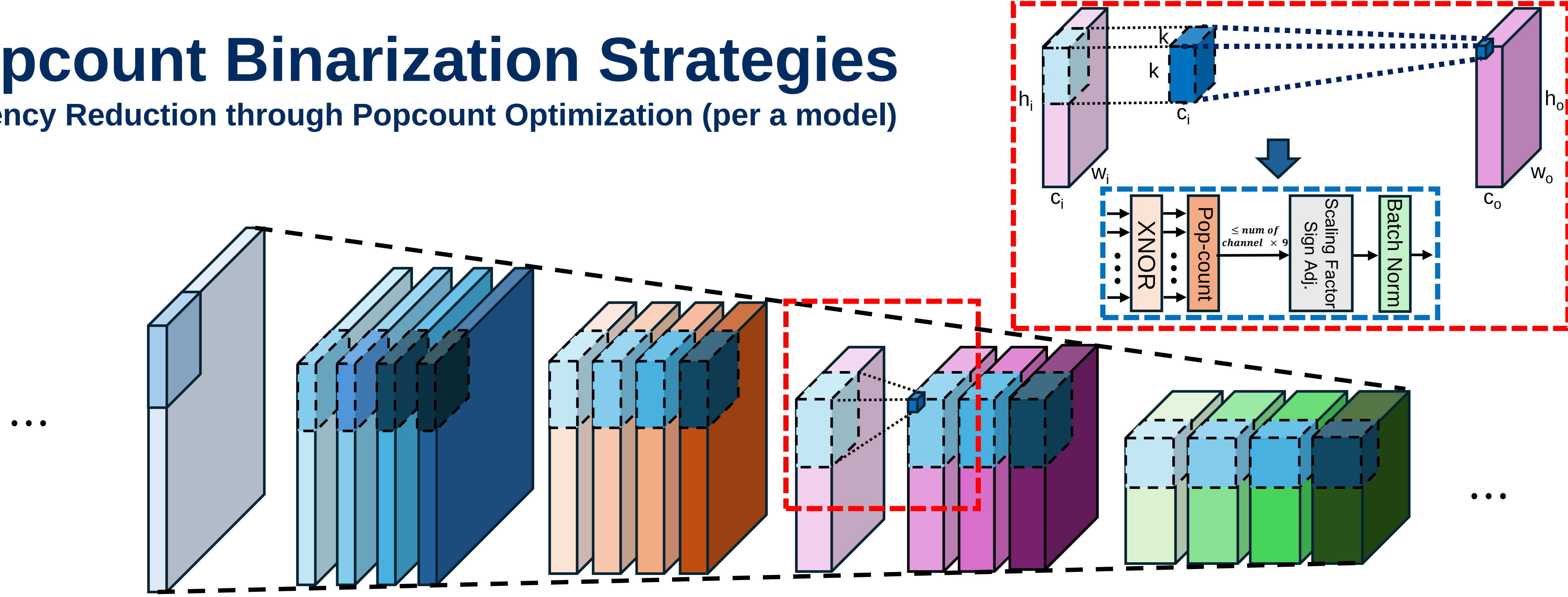
$\delta = \pm (channel\_num \times kernel^2 \div division\_num)$

: Inference

: Training

Popcount Binarization Strategies

- Latency Reduction through Popcount Optimization (per a model)



QAT-Popcount binarization ReActNet-18

Models	Operations	# Operations
ReActNet-18	Integer Multiplication & Bit shift	557,056 Integer Multiplications & Bit shifts
QAT-Popcount binarization ReActNet-18	None	None

Experiments

Datasets and Implementation Details

- Description of datasets used in the experiments and implementation details.

Optimization of Popcount Results

- Comparison and analysis of various techniques to optimize Popcount results.

Latency Efficiency Analysis

- Analysis of latency efficiency after applying Popcount optimization techniques.

Experiments

- Optimization of Popcount Results about PTQ-Popcount Binarization

Models	Top-1 Accuracy (%)	Top-5 Accuracy (%)
ReActNet-18	93.380	99.800
PTQ-Popcount binarization ReActNet-18	10.000	52.040
Bi-Real-18	88.770	98.250
PTQ-Popcount binarization Bi-Real-18	10.000	50.000

Experimental Settings

Dataset: CIFAR-10
Epoch: 128 for ReActNet-18, 256 for Bi-Real-18
Batch Size: 512

Experiments

- Optimization of Popcount Results about Simple QAT-Popcount Binarization

Models	Top-1 Accuracy (%)	Top-5 Accuracy (%)
ReActNet-18	93.380	99.800
Simple QAT-Popcount binarization ReActNet-18	84.930	99.250
Bi-Real-18	88.770	98.250
Simple QAT-Popcount binarization Bi-Real-18	30.070	79.690

Experiments

- Optimization of Popcount Results about QAT-Popcount Binarization depending on Division Num

Base Model	Division num	Top-1 Accuracy (%)	Top-5 Accuracy (%)
QAT-Popcount binarization ReActNet-18 (PopBin)	$channel\, num + \alpha$	92.150	99.640
	$(channel\, num \times kernel^2) + \alpha$	89.580	99.460
	$channel\, num \times \alpha$	92.510	99.640
	$(channel\, num \times kernel^2) \times \alpha$	92.160	99.660
	Min-Max Normalization $(channel\, num \times kernel^2)$	89.230	99.390

Experiments

- **Optimization of Popcount Results about QAT-Popcount Binarization**

Models	Top-1 Accuracy (%)	Top-5 Accuracy (%)
ReActNet-18	93.380	99.800
Simple QAT-Popcount binarization ResNet-18	84.930	99.250
QAT-Popcount binarization ReActNet-18 (PopBin)	92.510	99.640
Bi-Real-18	88.770	98.250
Simple QAT-Popcount binarization Bi-Real-18	30.070	79.690
QAT-Popcount binarization Bi-Real-18 (PopBin)	88.520	98.380

Discussion

Potential for Majority Voter Design

- Exploring the potential for hardware optimization using Majority Voter design to enhance performance.

Hierarchical and Approximate Majority Voter Design

- Analyzing the contribution of hierarchical and approximate Majority Voter designs to improving hardware efficiency.

Thank you