Deep Learning Optimization - Quantization

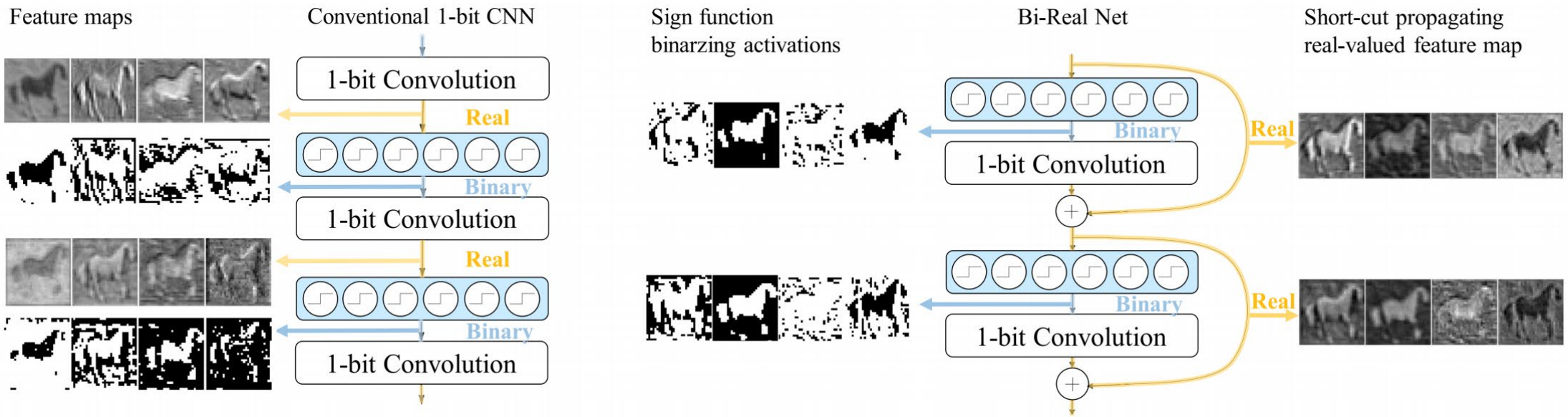
May 3, 2023

Eunhyeok Park

Bi-Real Net

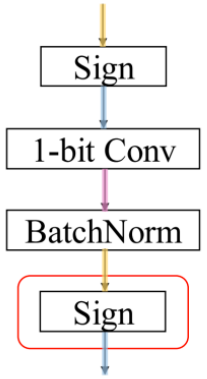
High-precision Identity Path

- The binary residual update is accumulated over the high-precision identity path

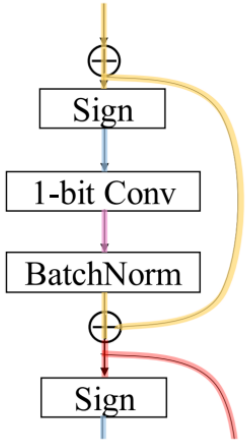


Representation Power of Mixed-precision Connection

- The representational capability of the Bi-Real net is significantly enhanced
- Ex) 32x32x3x3 binary convolution
  - Output range: -288 to 288

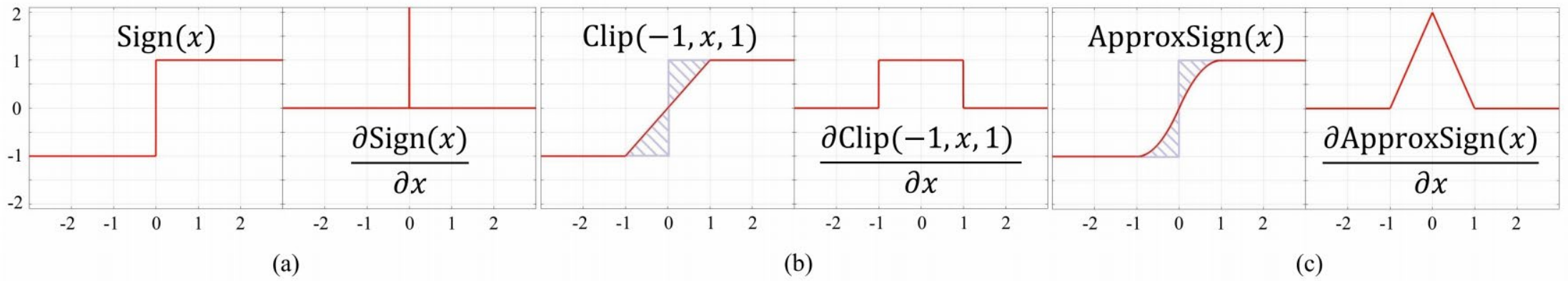


	Map size	Value range	Different value	$\mathbb{R}$
A_b^l	14x14x32	{-1, 1}	2	$2^{14x14x32}$
A_m^l	14x14x32	{-288,-286,...,288}	289	$289^{14x14x32}$
A_r^{l+1}	14x14x32	{a certain range}	289	$289^{14x14x32}$
A_b^{l+1}	14x14x32	{-1, 1}	2	$2^{14x14x32}$



	Map size	Value range	Different value	$\mathbb{R}$
A_r^l	14x14x32	{a certain range}	289	$289^{14x14x32}$
A_b^l	14x14x32	{-1, 1}	2	$2^{14x14x32}$
A_m^l	14x14x32	{-288,-286,...,288}	289	$289^{14x14x32}$
A_r^{l+1}	14x14x32	{a certain range}	289	$289^{14x14x32}$
A_{add}^{l+1}	14x14x32	{a certain range}	289*289	$289^{2x14x14x32}$

Binary Activation Approximation



$$\frac{\partial \mathcal{L}}{\partial \mathbf{A}_r^{l,t}} = \frac{\partial \mathcal{L}}{\partial \mathbf{A}_b^{l,t}} \frac{\partial \mathbf{A}_b^{l,t}}{\partial \mathbf{A}_r^{l,t}} = \frac{\partial \mathcal{L}}{\partial \mathbf{A}_b^{l,t}} \frac{\partial Sign(\mathbf{A}_r^{l,t})}{\partial \mathbf{A}_r^{l,t}} \approx \frac{\partial \mathcal{L}}{\partial \mathbf{A}_b^{l,t}} \frac{\partial F(\mathbf{A}_r^{l,t})}{\partial \mathbf{A}_r^{l,t}}$$

$$F(a_r) = \begin{cases} -1 & \text{if } a_r < -1 \\ 2a_r + a_r^2 & \text{if } -1 \leqslant a_r < 0 \\ 2a_r - a_r^2 & \text{if } 0 \leqslant a_r < 1 \\ 1 & \text{otherwise} \end{cases}, \quad \frac{\partial F(a_r)}{\partial a_r} = \begin{cases} 2 + 2a_r & \text{if } -1 \leqslant a_r < 0 \\ 2 - 2a_r & \text{if } 0 \leqslant a_r < 1 \\ 0 & \text{otherwise} \end{cases},$$

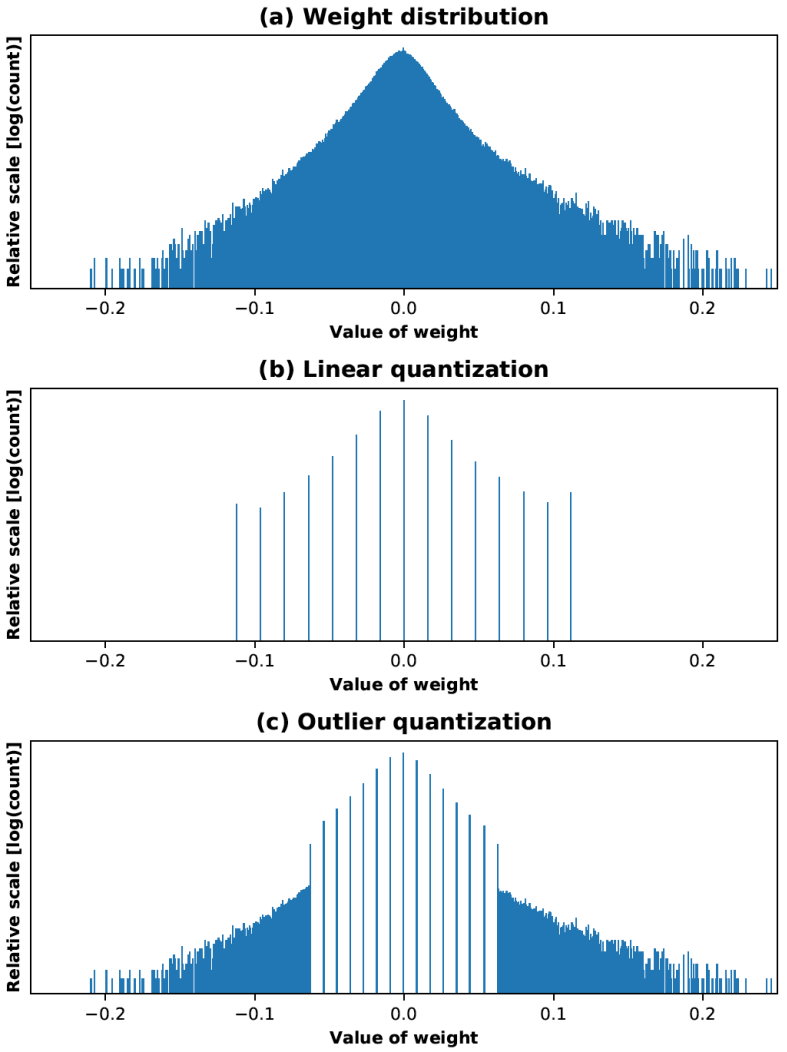
Results

Initialization	Weight update	Activation backward	Bi-Real-18 top-1	Bi-Real-18 top-5	Res-18 top-1	Res-18 top-5	Plain-18 top-1	Plain-18 top-5	Bi-Real-34 top-1	Bi-Real-34 top-5	Res-34 top-1	Res-34 top-5	Plain-34 top-1	Plain-34 top-5
ReLU	Original	Original	32.9	56.7	27.8	50.5	3.3	9.5	53.1	76.9	27.5	49.9	1.4	4.8
		Proposed	36.8	60.8	32.2	56.0	4.7	13.7	58.0	81.0	33.9	57.9	1.6	5.3
	Proposed	Original	40.5	65.1	33.9	58.1	4.3	12.2	59.9	82.0	33.6	57.9	1.8	6.1
		Proposed	47.5	71.9	41.6	66.4	8.5	21.5	61.4	83.3	47.5	72.0	2.1	6.8
	Real-valued Net		68.5	88.3	67.8	87.8	67.5	87.5	70.4	89.3	69.1	88.3	66.8	86.8
Clip	Original	Original	37.4	62.4	32.8	56.7	3.2	9.4	55.9	79.1	35.0	59.2	2.2	6.9
		Proposed	38.1	62.7	34.3	58.4	4.9	14.3	58.1	81.0	38.2	62.6	2.3	7.5
	Proposed	Original	53.6	77.5	42.4	67.3	6.7	17.1	60.8	82.9	43.9	68.7	2.5	7.9
		Proposed	**56.4**	**79.5**	45.7	70.3	12.1	27.7	**62.2**	**83.9**	49.0	73.6	2.6	8.3
	Real-valued Net		68.0	88.1	67.5	87.6	64.2	85.3	69.7	89.1	67.9	87.8	57.1	79.9
Full-precision original ResNet[5]					69.3	89.2					73.3	91.3		

Outlier-aware Quantization

Basic Idea: Outlier-aware Quantization

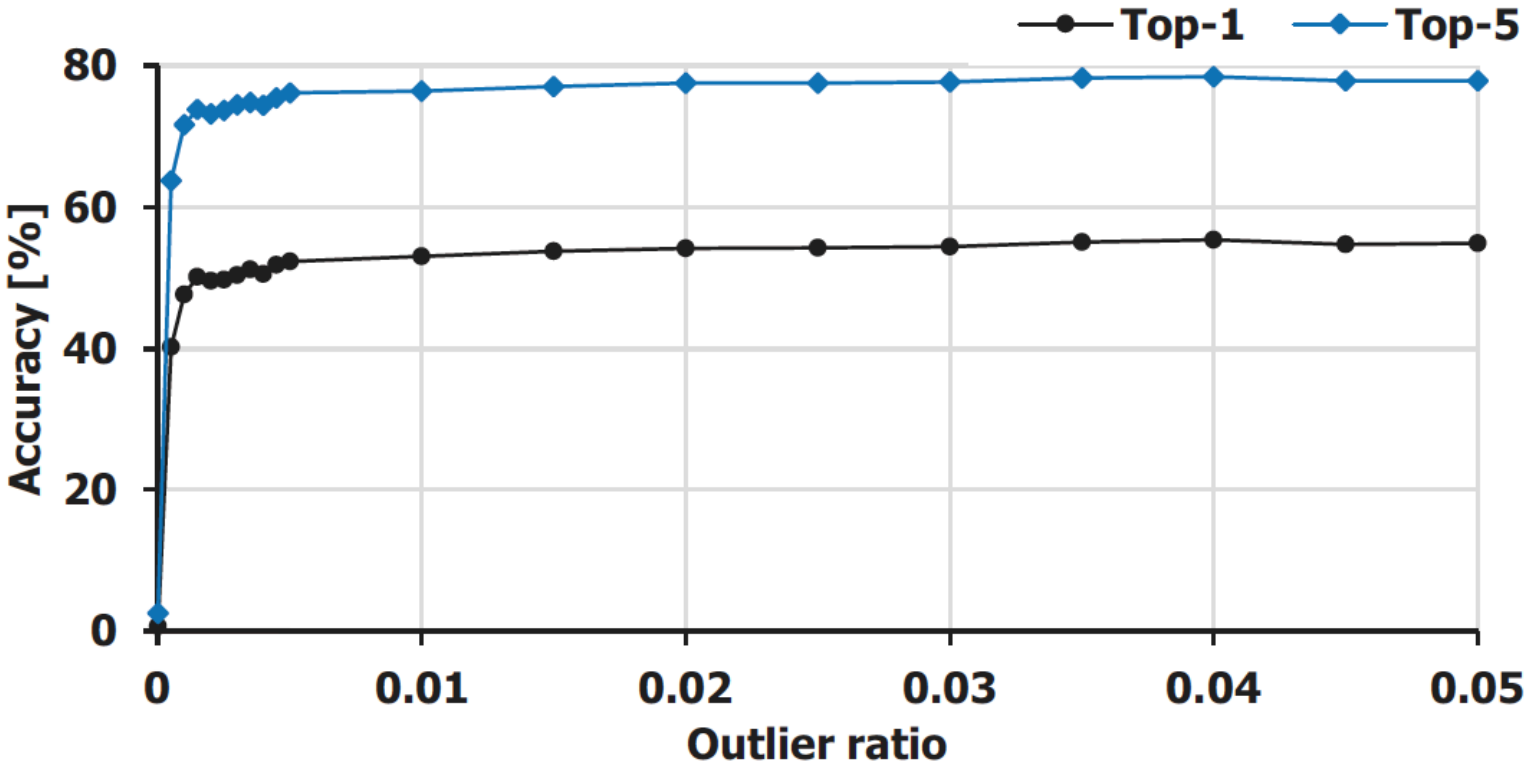
- Observation
  - Reduced precision suffers from quantization error mainly due to outliers
- Proposed idea
  - Handle outliers (e.g., 1~3% of total data) separately from the other data
- Expected effects
  - Smaller quantization error for all the data
  - Outlier → high precision, e.g., 16 bits, at no quantization error
  - Majority of data → reduced precision with narrower region, i.e., smaller quantization error

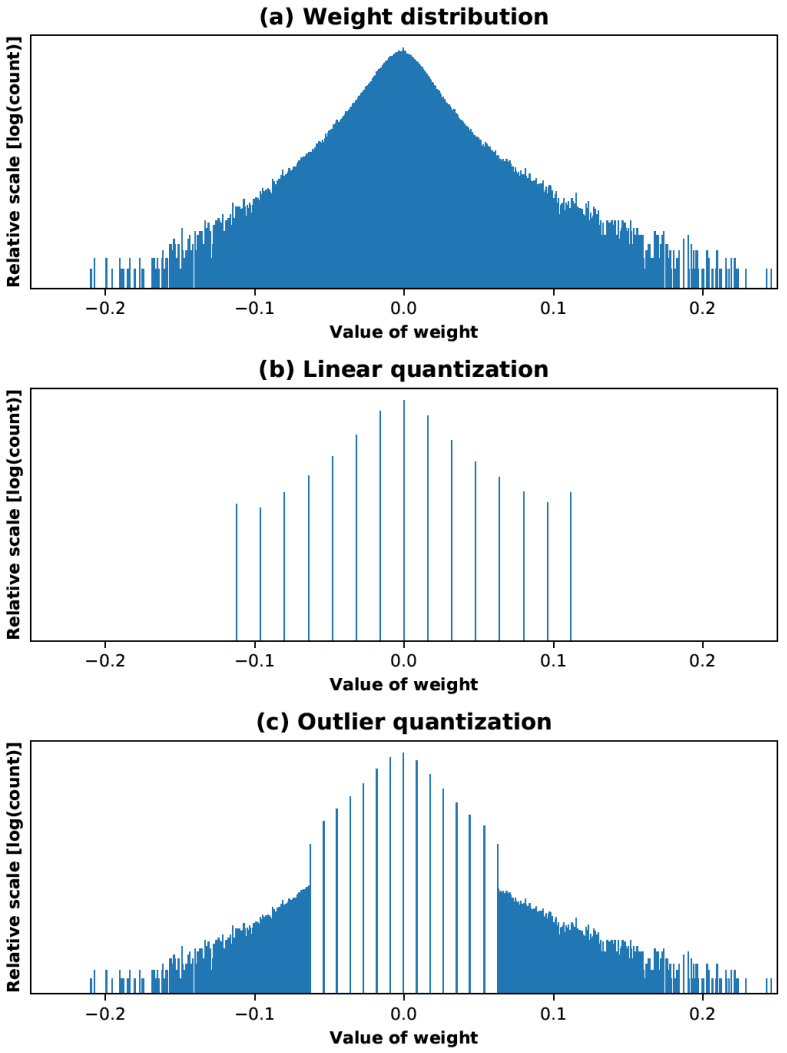


Weights in 2nd CONV of AlexNet

Feasibility Test: AlexNet

- 4-bit quantization with outliers of 0.5% already gives good accuracy
- Outliers of 3.5% lose only <1% accuracy

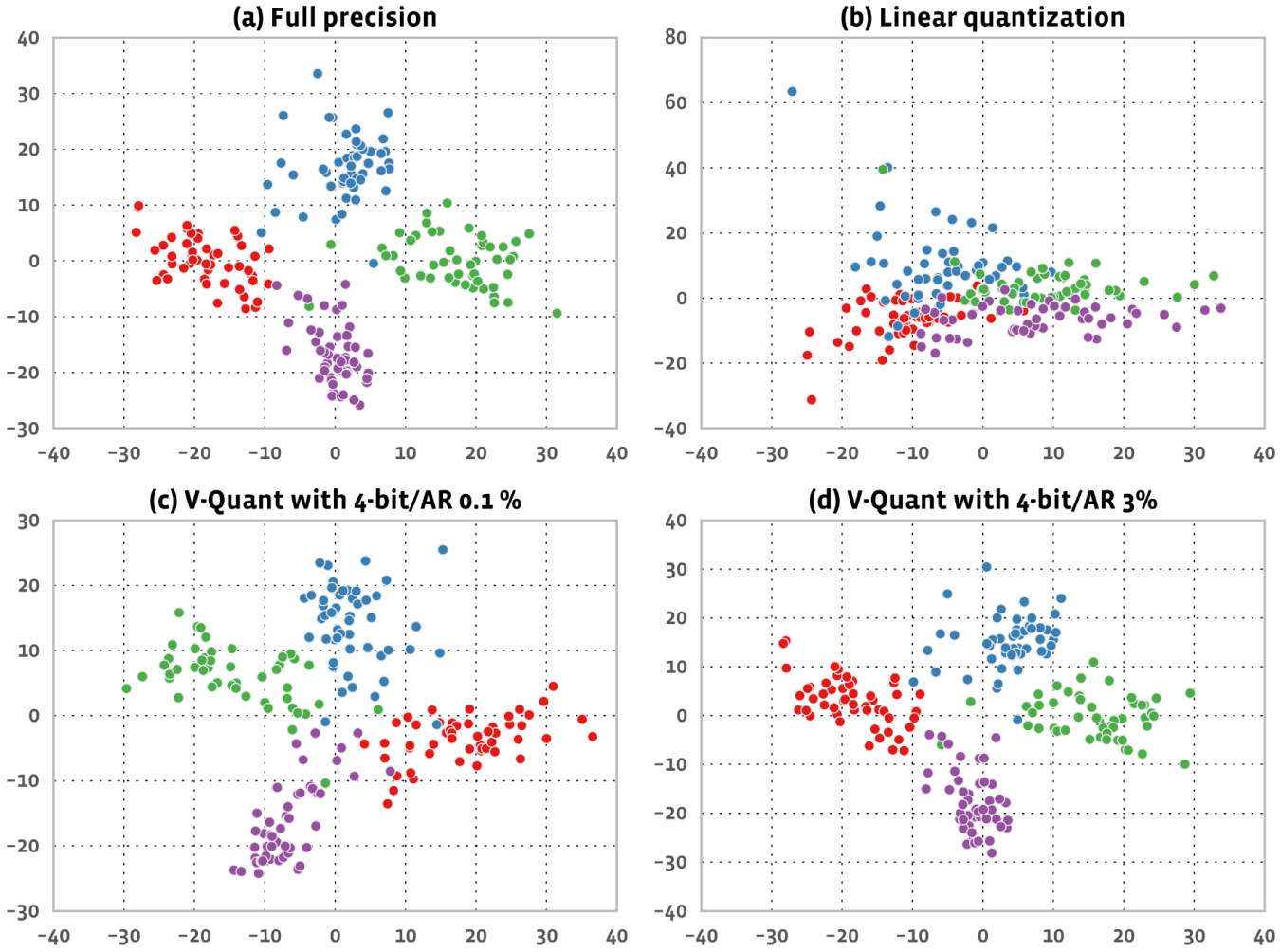




Weights in 2nd CONV of AlexNet

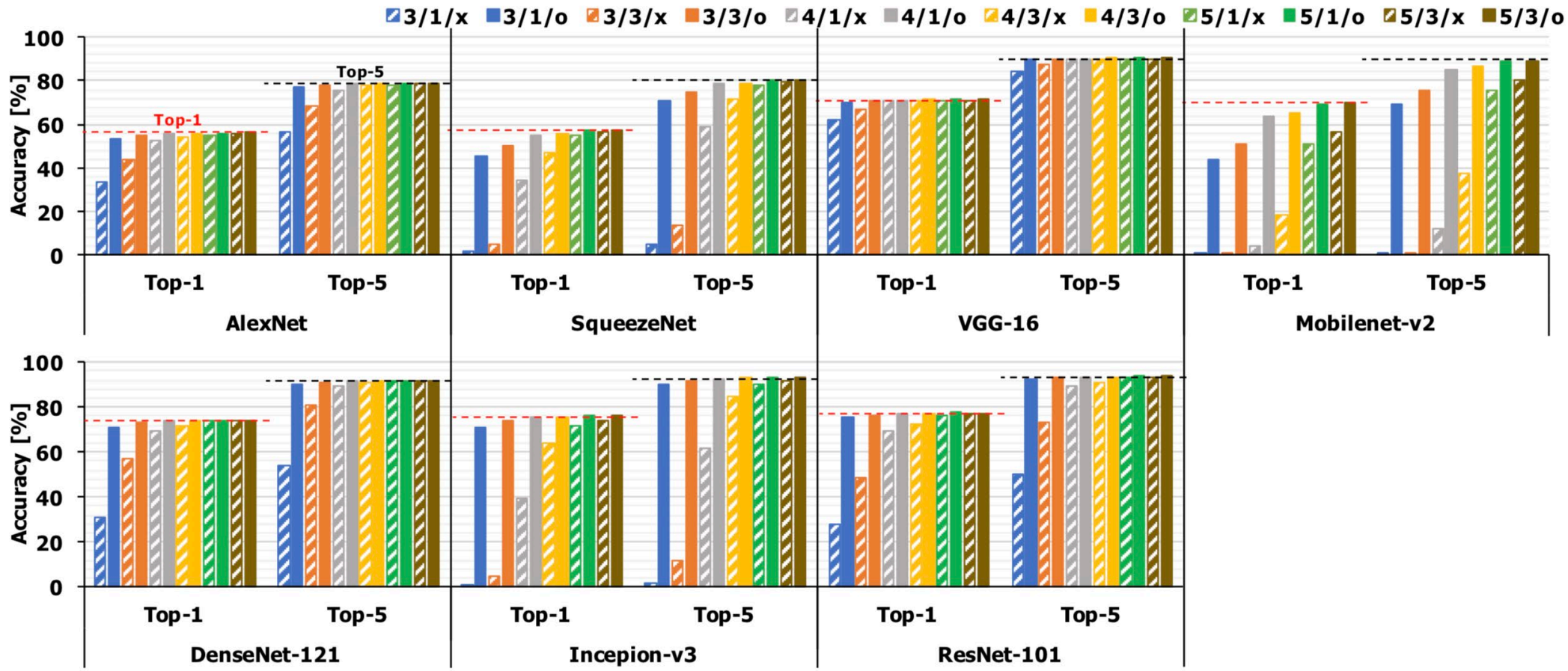
Effects of Outliers

- PCA analysis
  - Full precision vs. 4 bit
- Conv5 in AlexNet
- Four groups of classes start to be separated even with 0.1% of outliers



OLQuant for Inference (Fine-tuning)

- 4 (5) bits for 99% weights/acts (1% 16b) give <1% top-1 accuracy loss

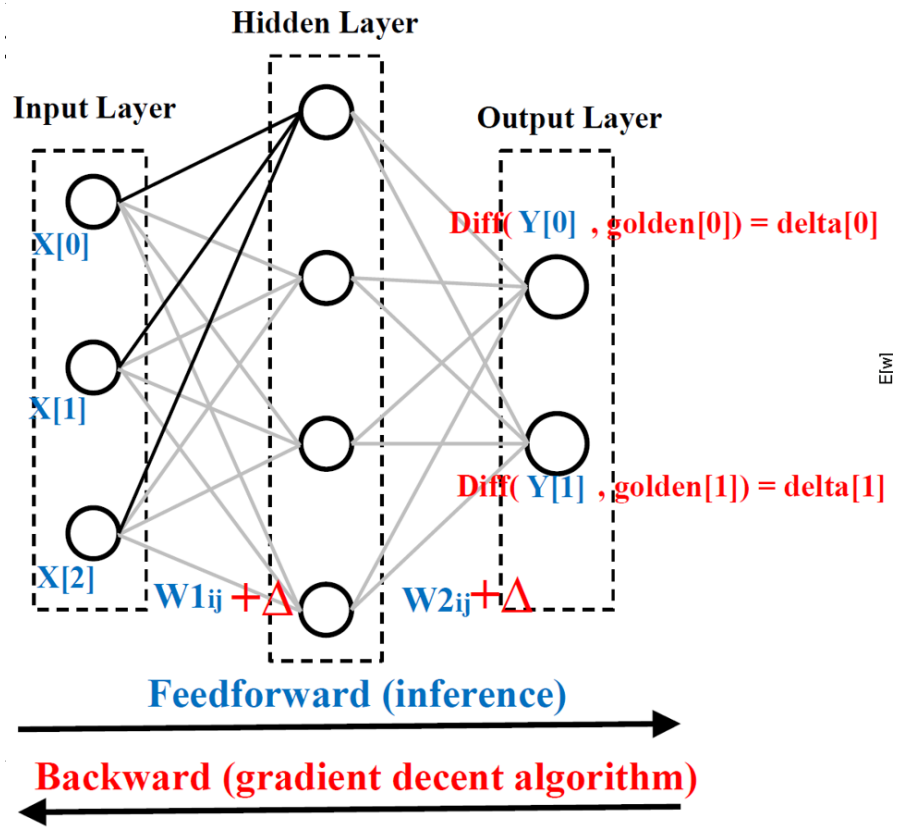


OLQuant for Acceleration

- Facebook case study reports 1.7X speedup on Intel CPU
  - Current Intel CPU: 16-bit x/32-bit + or 8-bit x/16-bit + (2x faster)
  - Problem?
    - Overflow when using 8-bit x/16-bit +
  - Dense 8-bit mult/16-bit accum with sparse outlier 16-bit mult/32-bit accum
  - “Deep Learning Inference in Facebook Data Centers: Characterization, Performance Optimizations and Hardware Implications,” arXiv:1811.09886, 2018.

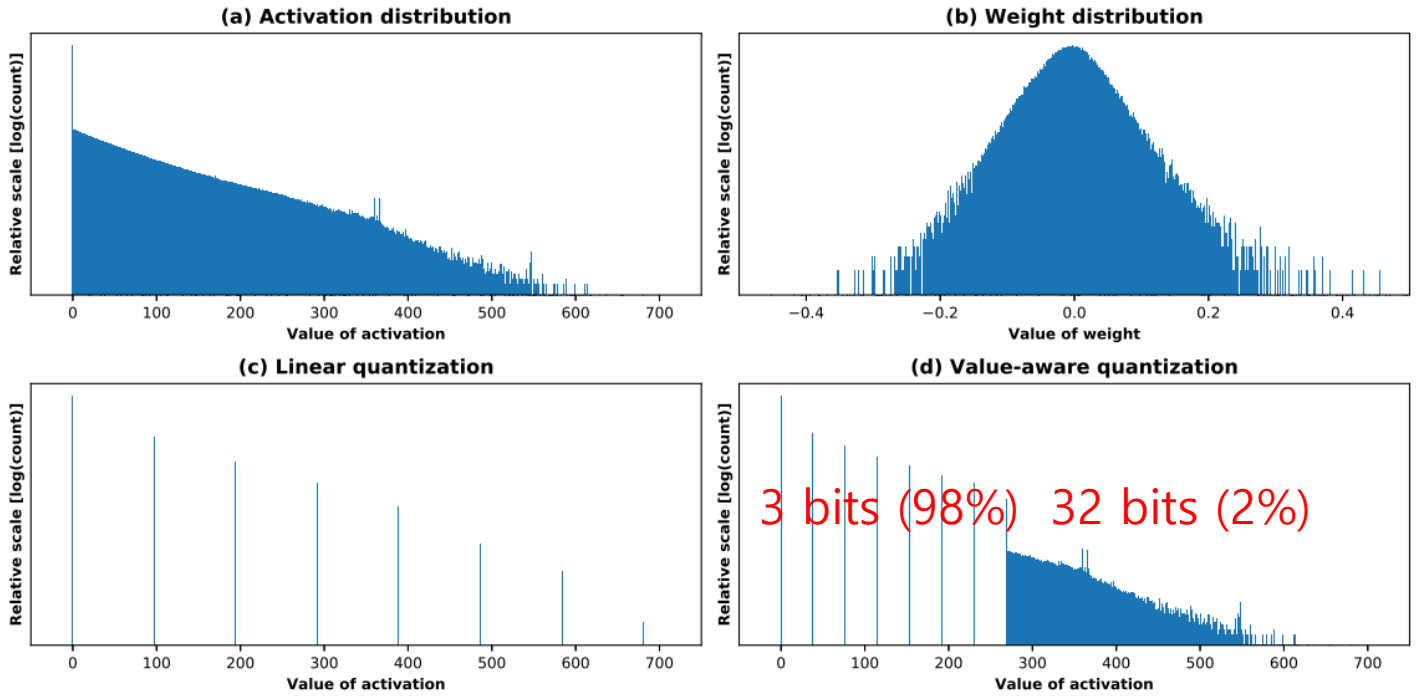
Large Memory is Required for Training

- Large memory is required when training large neural networks with large batches
- Intermediate activations are needed to be stored for the back-propagation
  - Weight size is much smaller than activation in CNN
- Example
  - 1 layer typically needs ~1MB of activation
  - 100-layer network needs to keep ~100MB of activation
  - 1k batch requires ~100GB of memory only for activation
  - E.g., DGX-1 (128GB) supports 2k batch for AlexNet



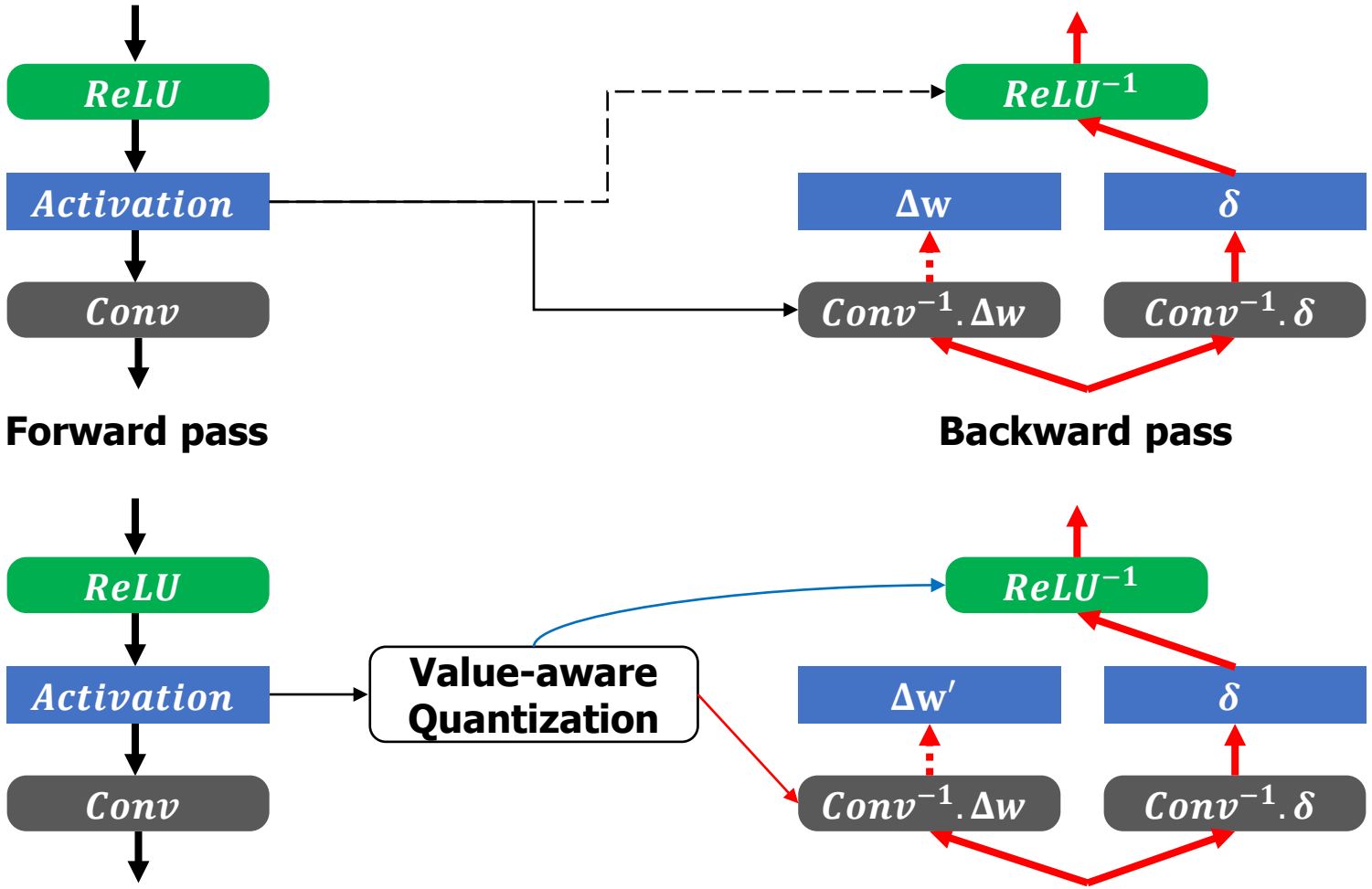
OLQuant for Training

- 9X reduction in stored activations for ResNet-50 training
  - 3 bits (98%) and 32 bits (2%) for activations
- Forward/backward passes do not change except weight update

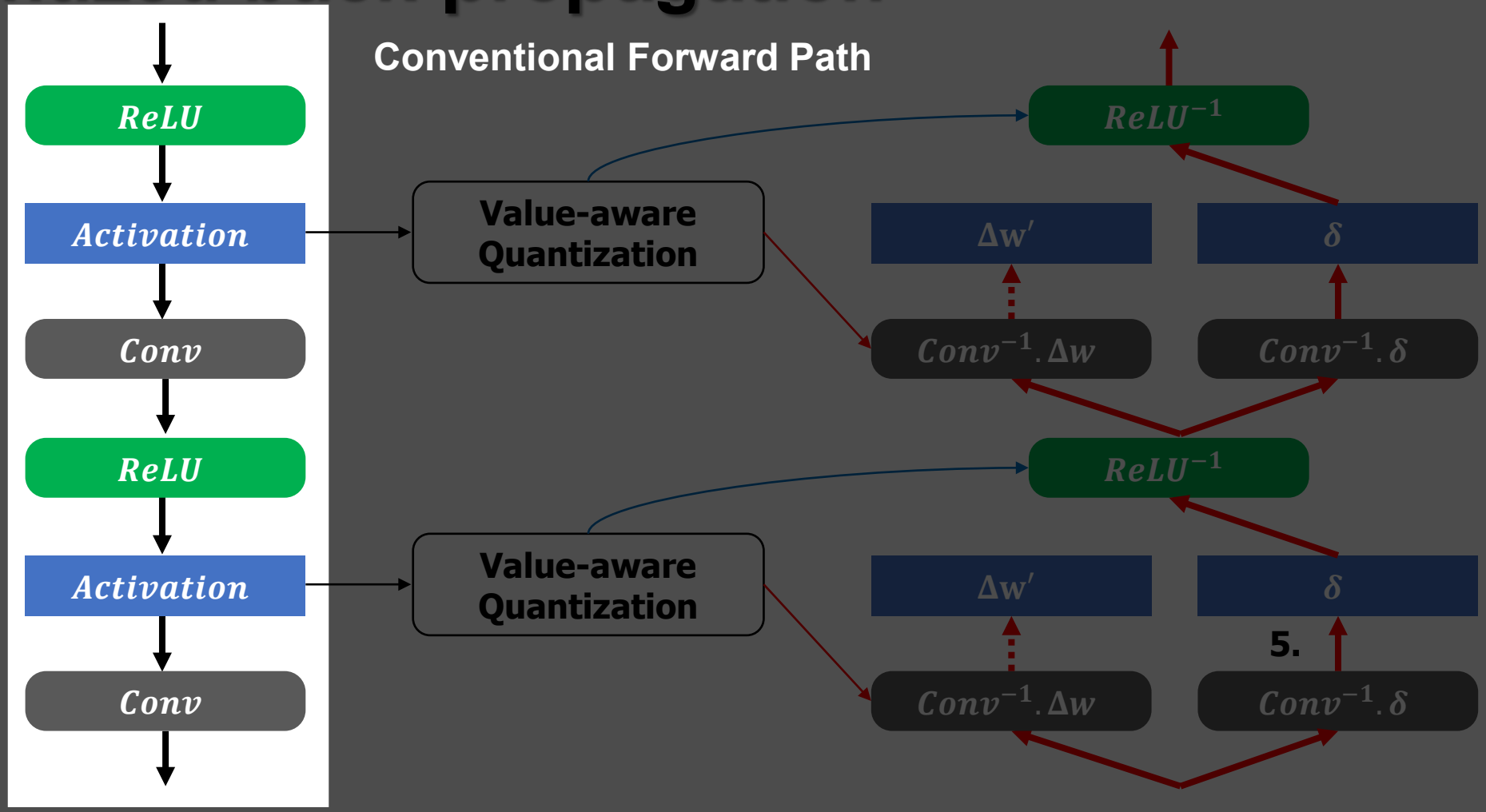


Quantized back-propagation

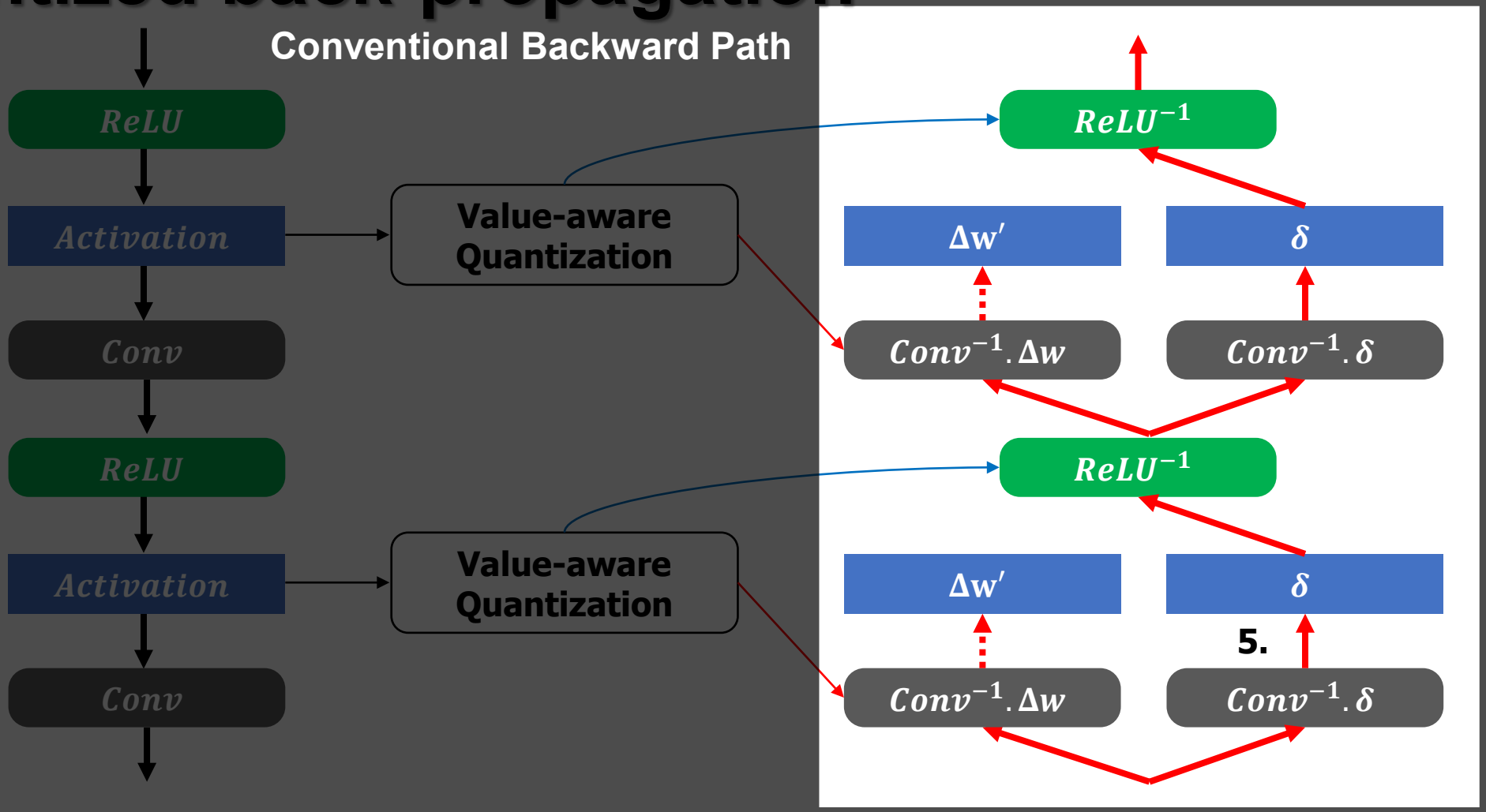
Idea: Store Compressed Activation based on OLQuant



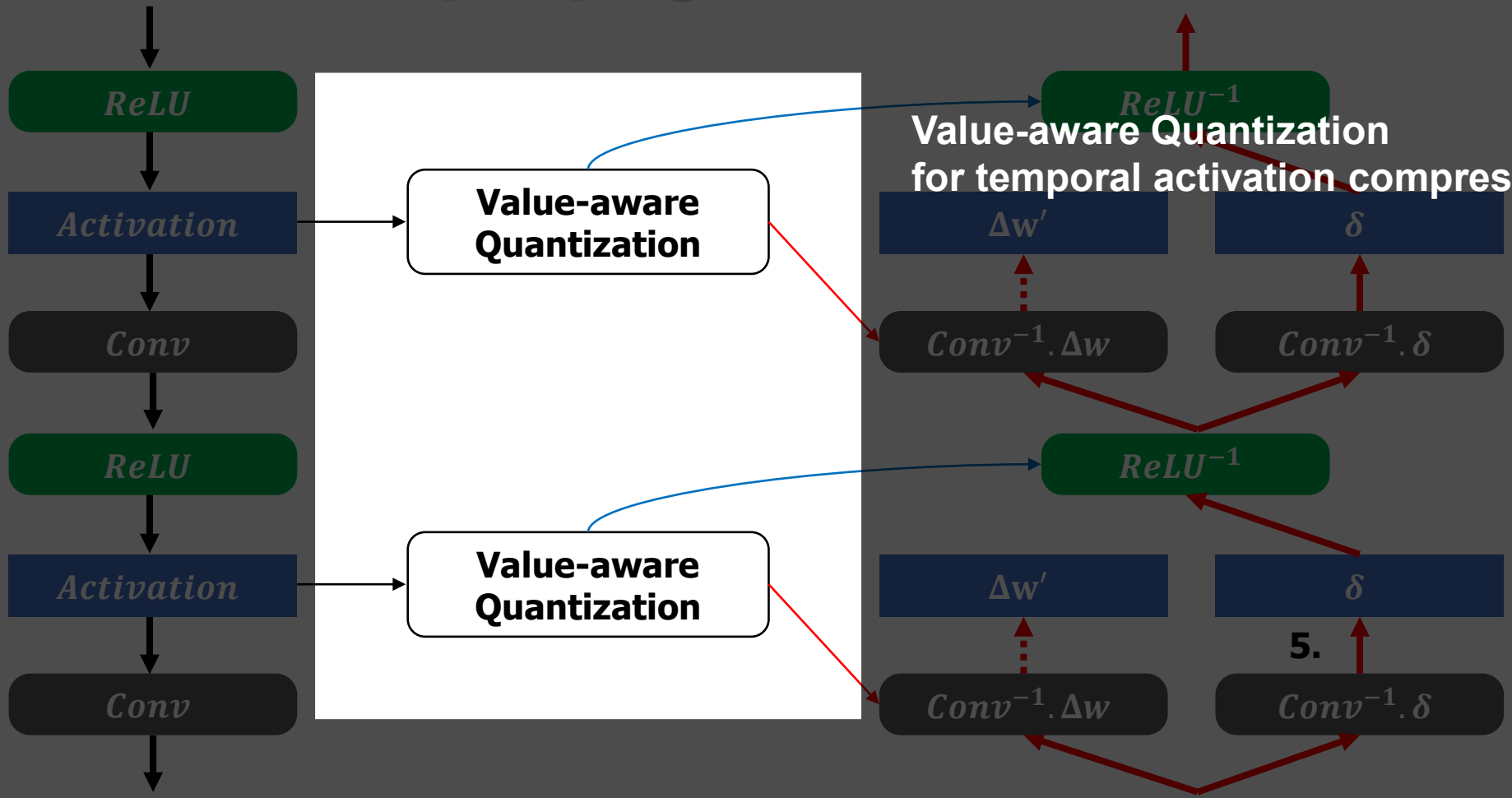
Quantized back-propagation



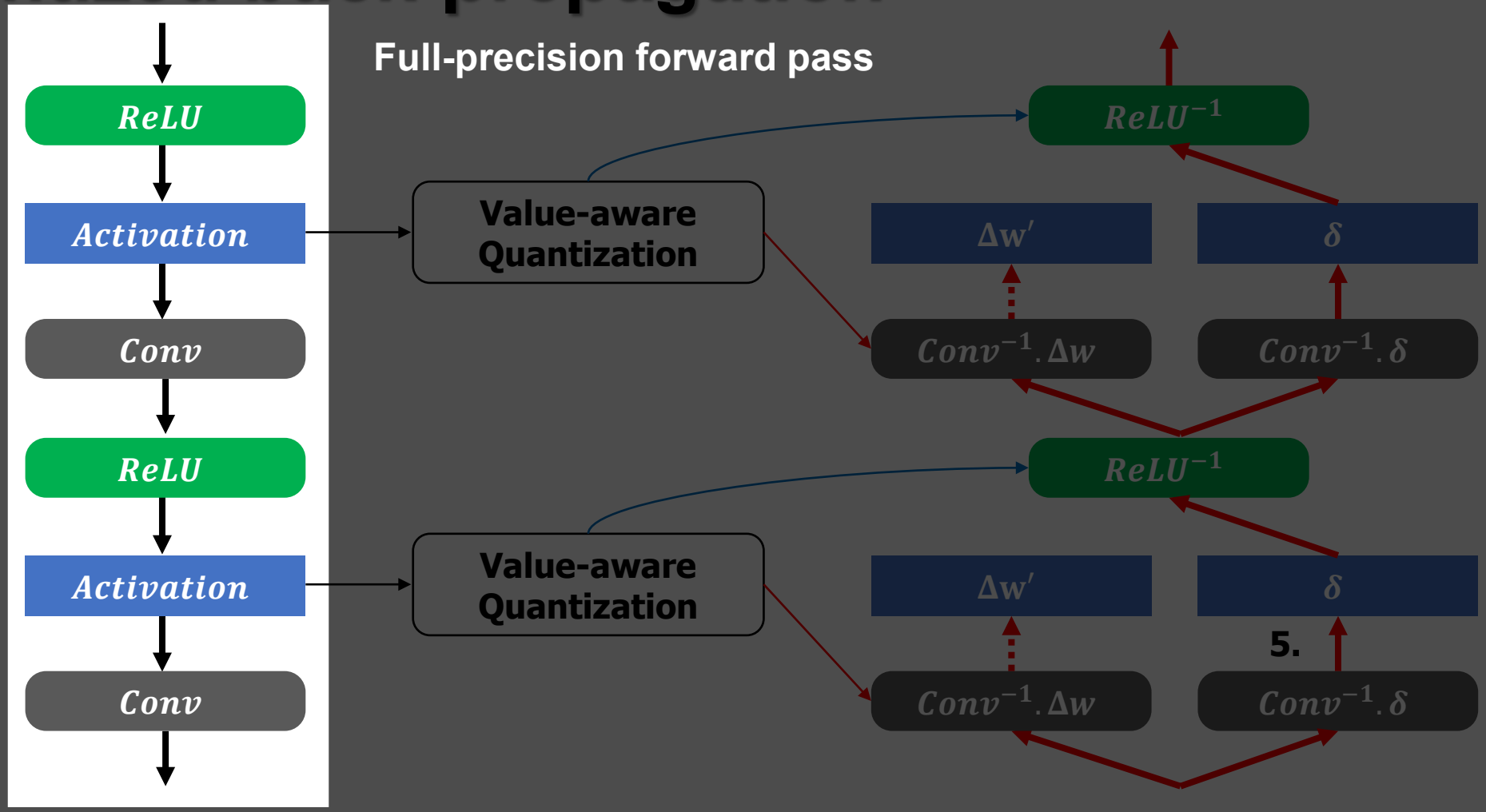
Quantized back-propagation



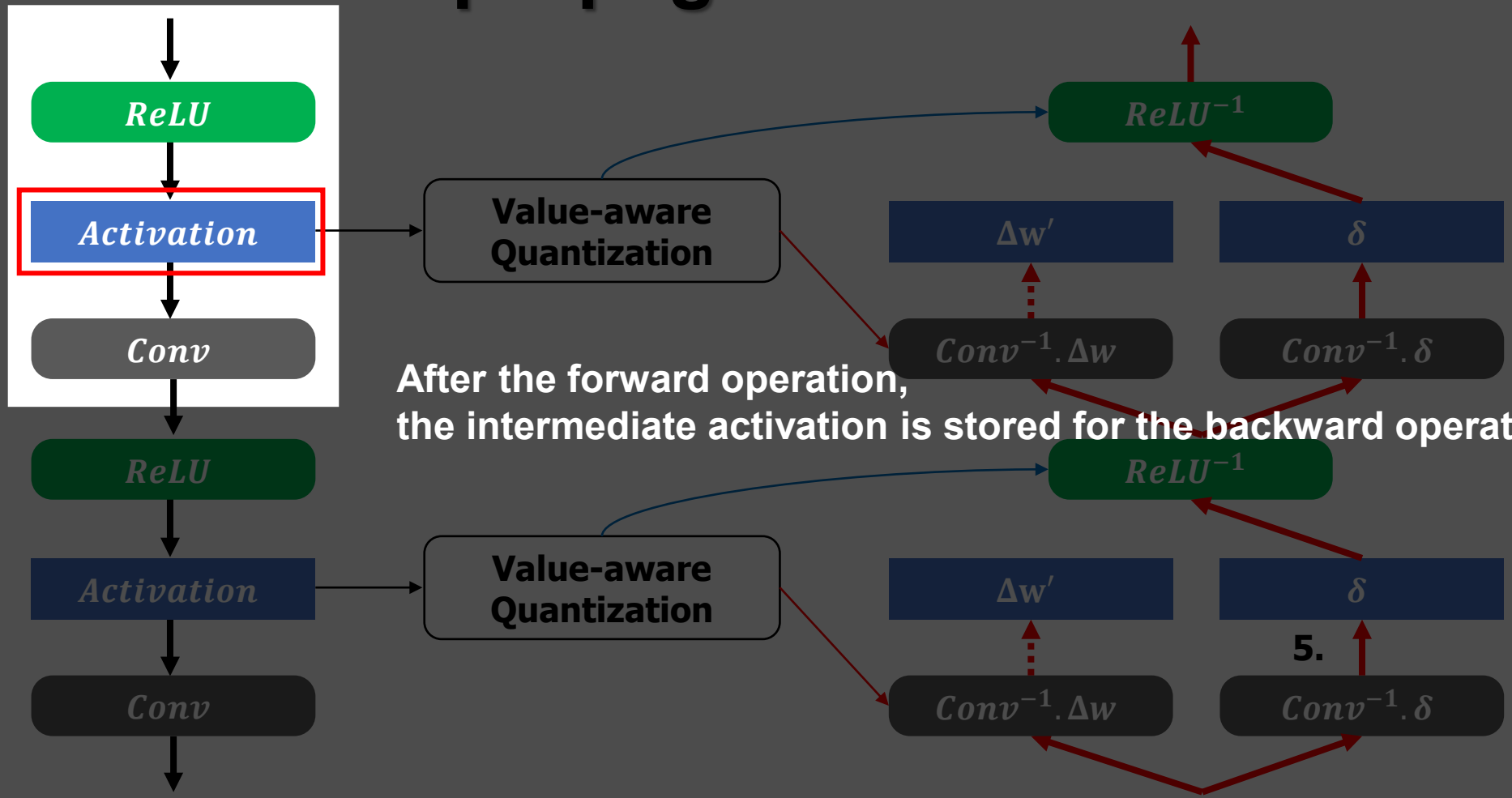
Quantized back-propagation



Quantized back-propagation

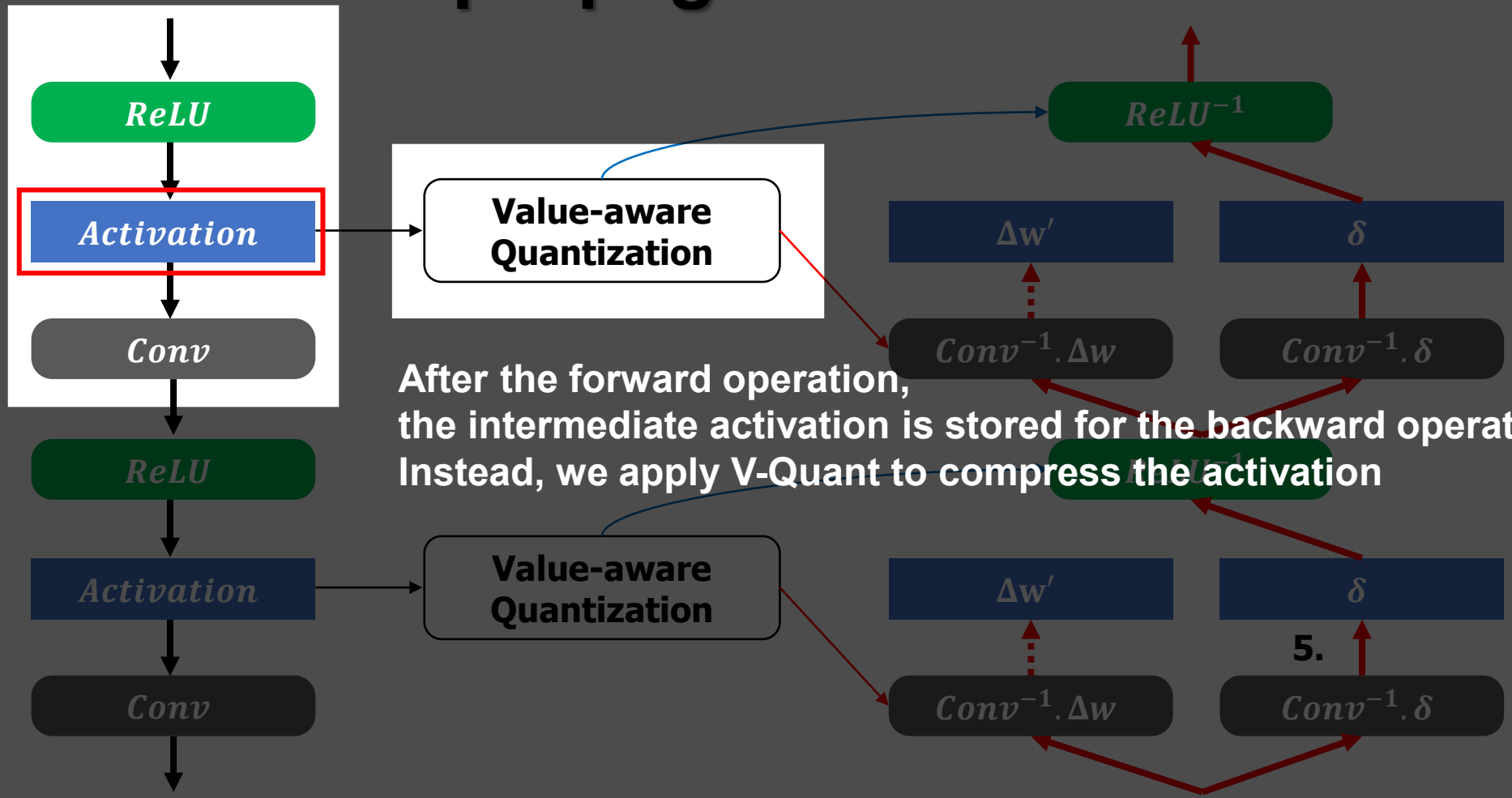


Quantized back-propagation



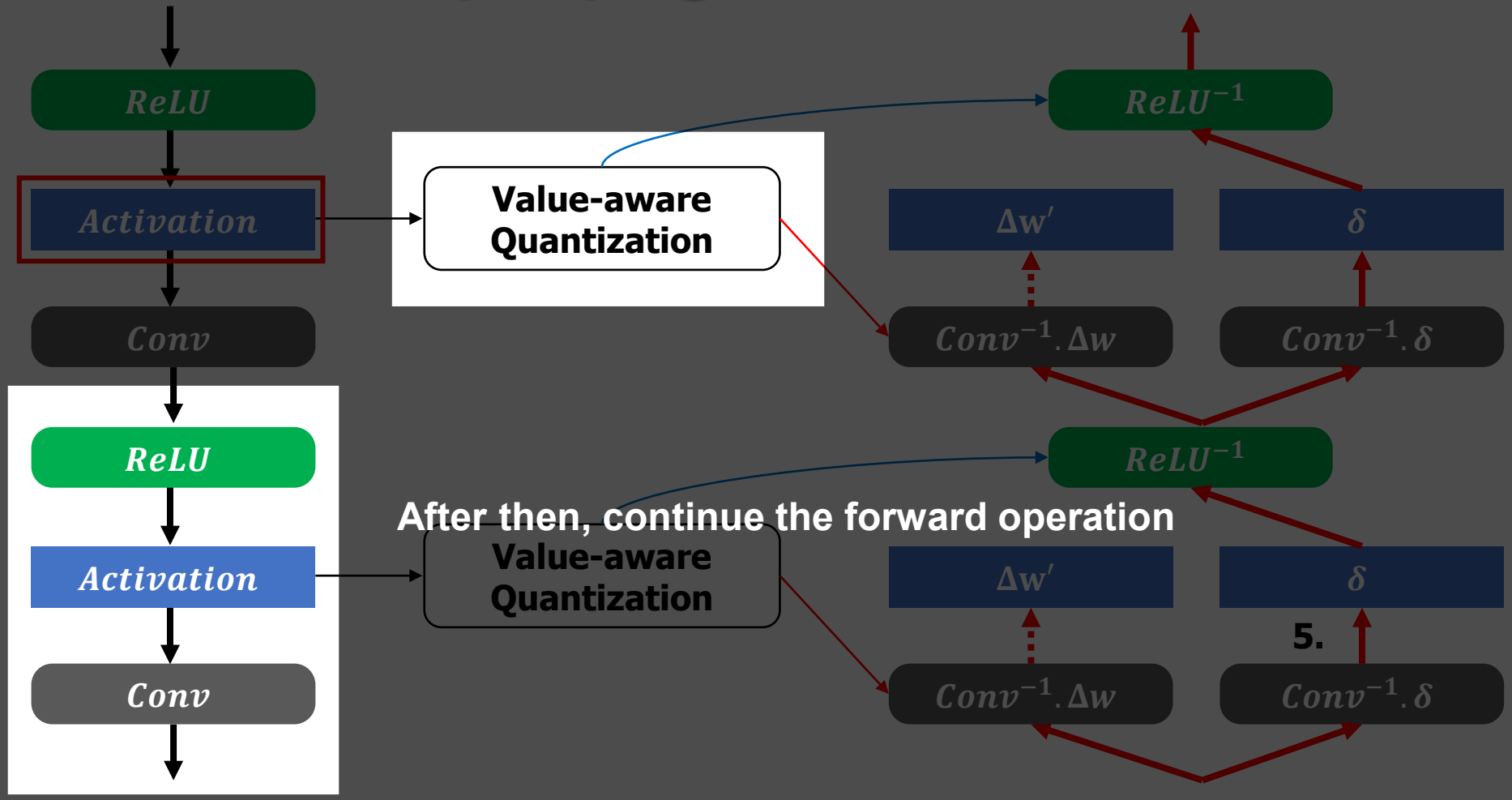
After the forward operation,
the intermediate activation is stored for the backward operation

Quantized back-propagation

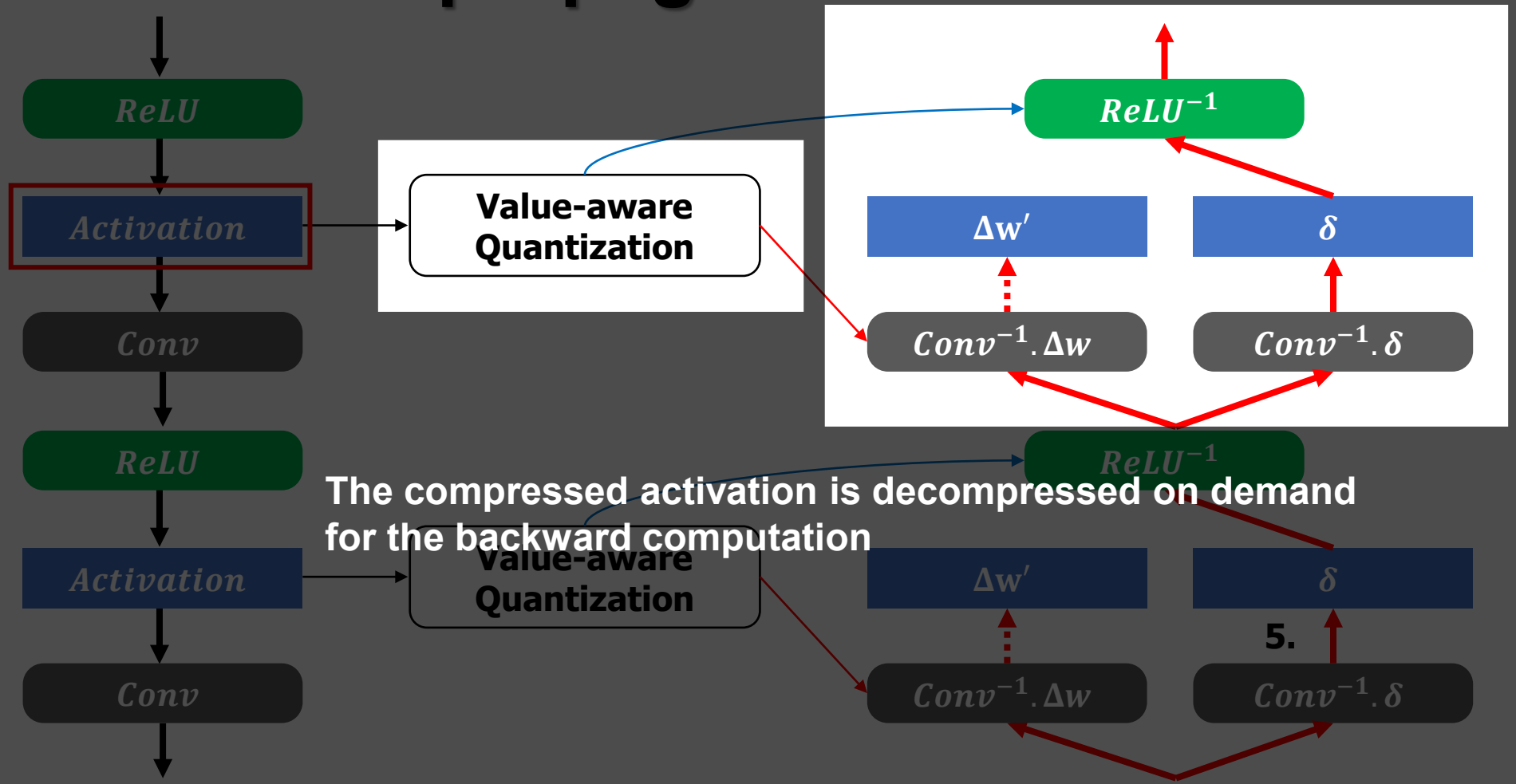


After the forward operation,
the intermediate activation is stored for the backward operation
Instead, we apply V-Quant to compress the activation

Quantized back-propagation

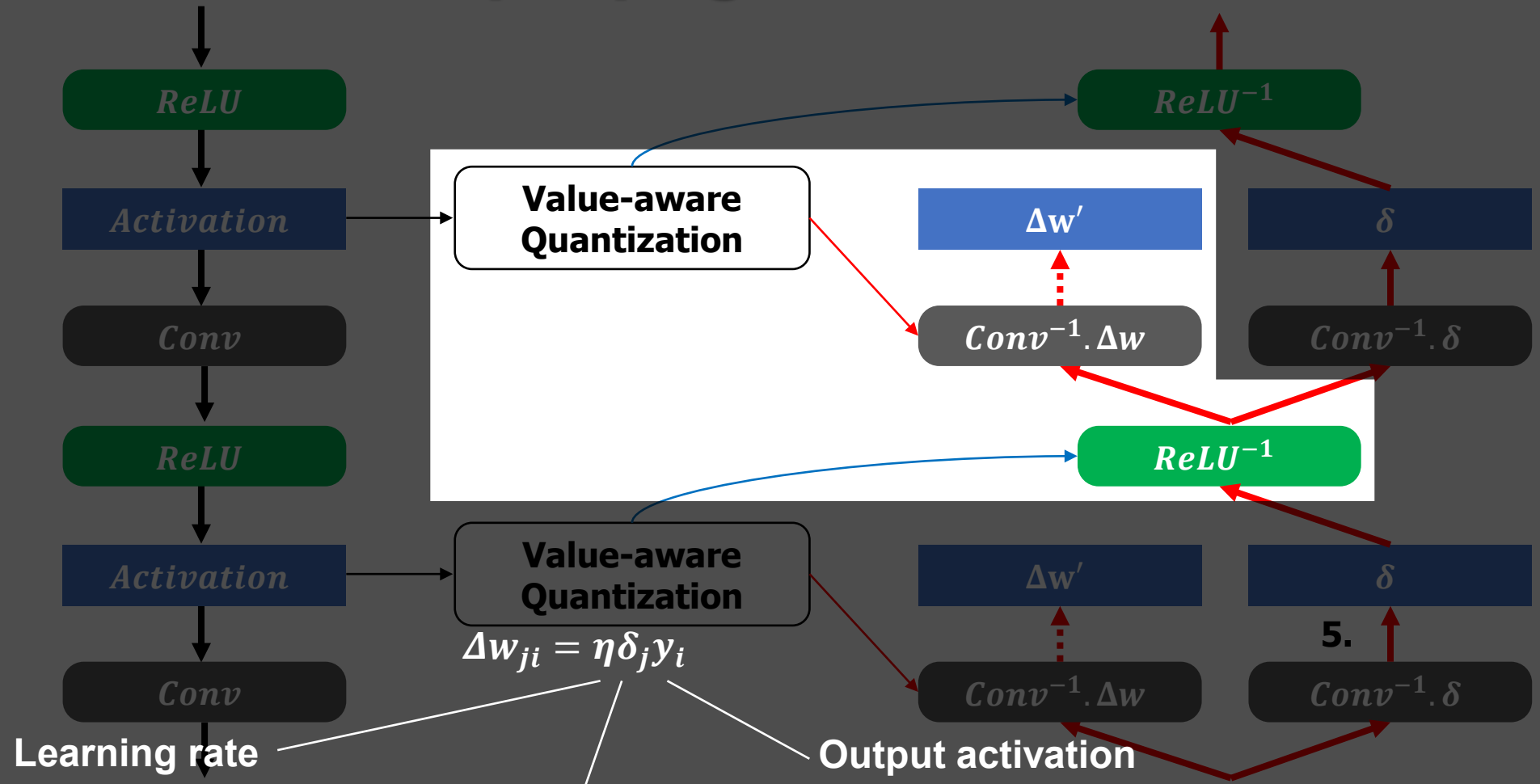


Quantized back-propagation

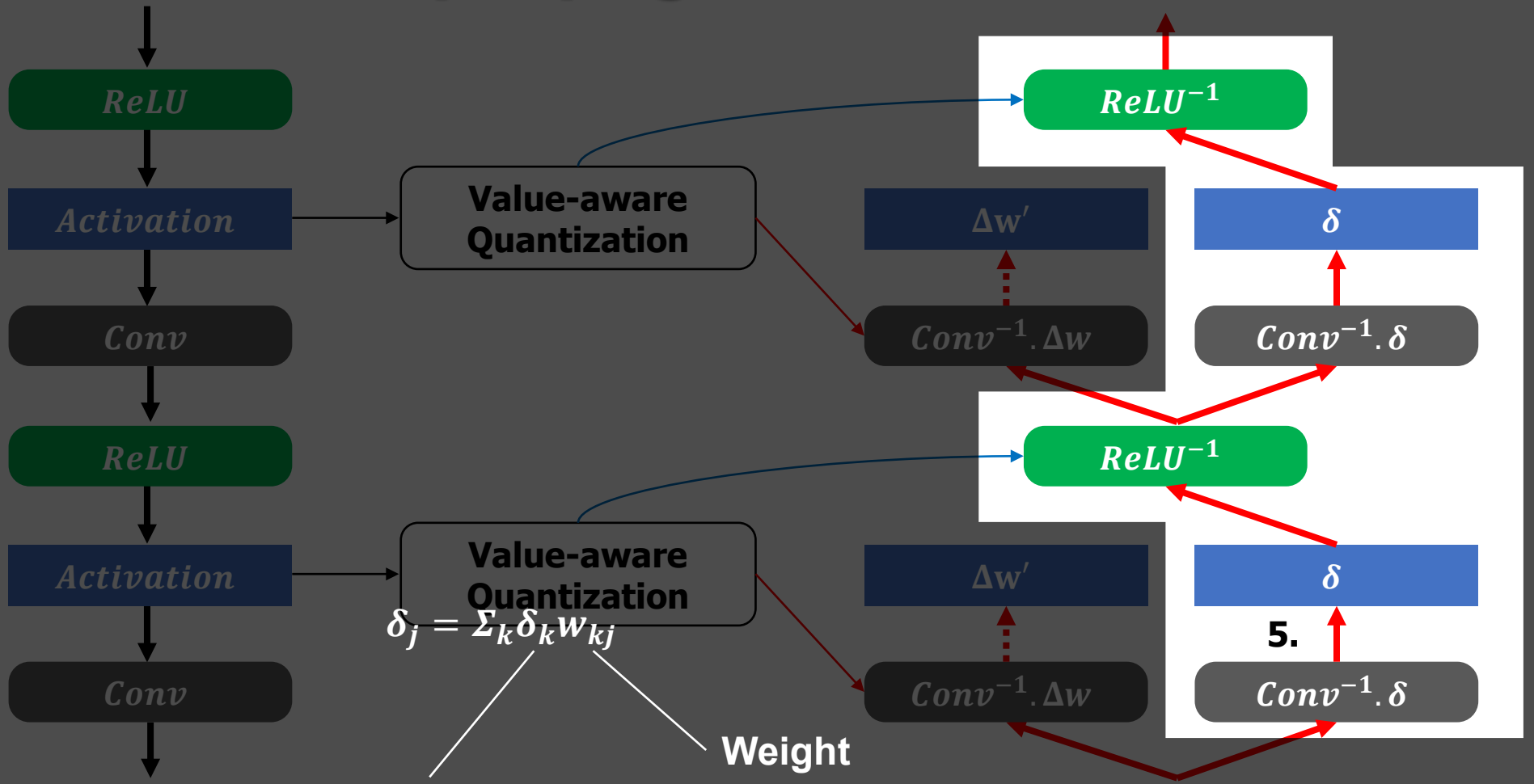


The compressed activation is decompressed on demand for the backward computation

Quantized back-propagation



Quantized back-propagation



- Gradient is the same as full-precision

ResNet-50 Training: 98% 3 bits and 2% 32 bits

AR [%]	0	1	2	3	4	5
1-bit	5.302 / 15.228	74.510 / 92.048	75.172 / 92.500	75.214 / 92.482	75.698 / 92.656	75.568 / 92.662
2-bit	65.754 / 86.718	75.652 / 92.658	75.638 / 92.702	75.660 / 92.512	75.338 / 92.660	75.576 / 92.615
3-bit	75.486 / 92.608	75.708 / 92.592	**75.920 / 92.858**	75.930 / 92.964	75.892 / 92.938	75.734 / 92.630
4-bit	75.700 / 92.750	75.784 / 92.670	75.880 / 92.926	75.790 / 92.712	75.846 / 92.694	75.916 / 92.858

5-bit with AR 0 %	75.600 / 92.610	**6-bit with AR 0 %**	75.922 / 92.832
7-bit with AR 0 %	75.887 / 92.792	**8-bit with AR 0 %**	75.670 / 92.846

- Floating-point accuracy (top-1/5): 75.916 / 92.904 %
- 6.1X memory reduction in activation
  - + LZ compression of 3-bit data → 9X reduction

Memory Cost Reduction in Training

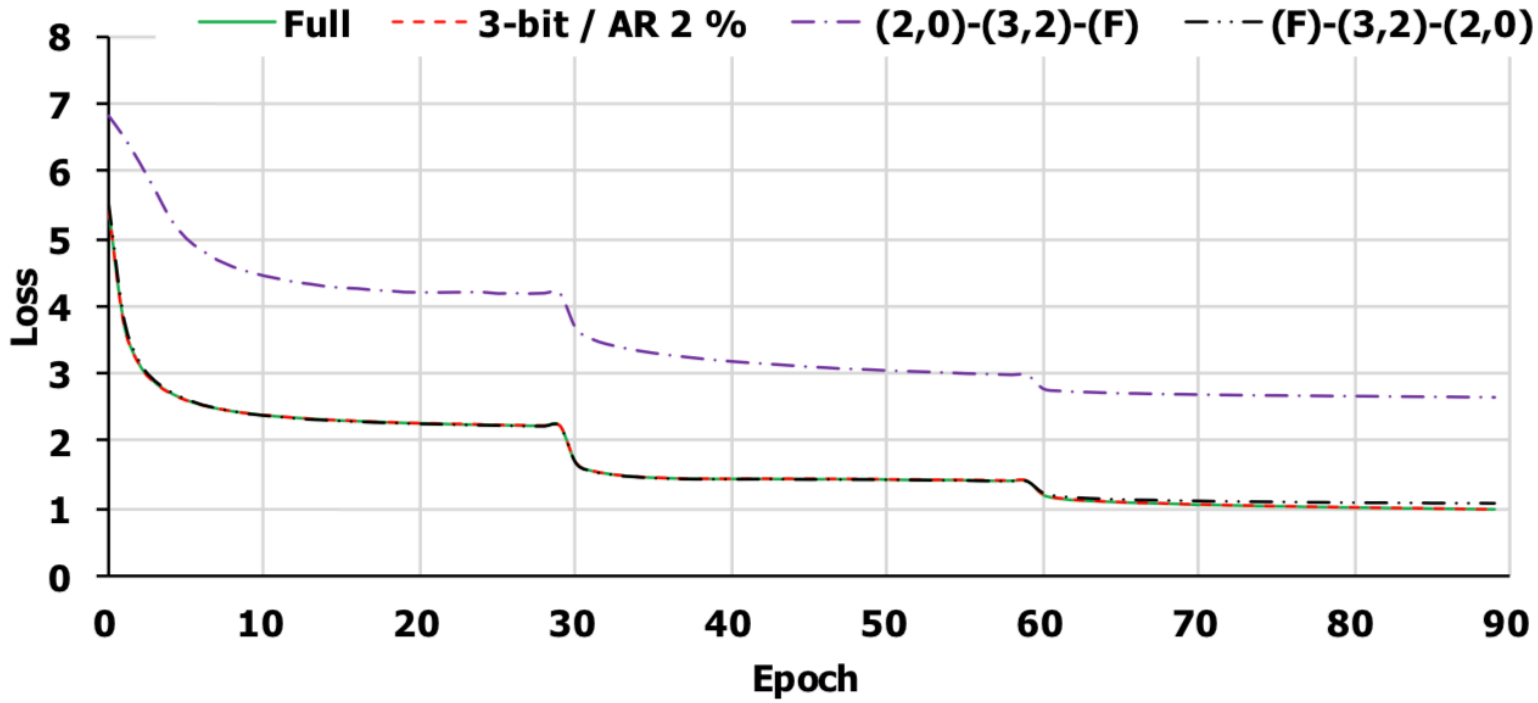
	AlexNet	ResNet-18	SqueezeNet-1.1	MobileNet-v2	VGG-16	Inception-v3	ResNet-152	DenseNet-201
Full	56.354 / 79.020	69.908 / 89.384	58.672 / 81.052	70.104 / 89.736	71.862 / 90.484	74.194 / 91.920	77.954 / 94.024	77.418 / 93.586
3-bit 2%	56.142 / 78.986	69.920 / 89.230	58.528 / 80.942	70.116 / 89.764	71.744 / 90.462	74.140 / 91.916	77.758 / 93.894	77.276 / 93.442
8-bit 0%	56.238 / 78.948	70.010 / 89.276	58.750 / 81.290	70.294 / 89.638	71.774 / 90.660	74.224 / 92.084	78.354 / 93.948	77.320 / 93.508

	AlexNet	ResNet-18	SqueezeNet-1.1	MobileNet-v2	ResNet-50	VGG-16	Inception-v3	ResNet-152	DenseNet-201
Full	0.35	1.86	1.58	7.34	9.27	9.30	9.75	20.99	24.53
Chen et al. [2]	x	0.98 (52.1 %)	1.05 (66.9 %)	4.21 (52.1 %)	3.70 (39.9 %)	x	3.87 (39.8 %)	5.29 (25.2 %)	6.62 (27.0 %)
(2,0)	0.23 (66.4 %)	0.42 (22.6 %)	0.59 (37.5 %)	0.74 (10.0 %)	1.22 (13.2 %)	3.65 (39.2 %)	1.16 (11.9 %)	1.64 (7.78 %)	2.09 (8.51 %)
(3,0)	0.23 (67.8 %)	0.46 (24.3 %)	0.61 (38.8 %)	0.84 (11.4 %)	1.34 (14.5 %)	3.75 (40.3 %)	1.43 (14.8 %)	2.27 (10.8 %)	2.85 (11.6 %)
(3,2)	0.24 (69.5 %)	0.50 (26.5 %)	0.64 (40.4 %)	1.13 (15.4 %)	1.52 (16.4 %)	3.88 (41.7 %)	1.79 (18.4 %)	3.09 (14.7 %)	3.83 (15.6 %)

Activation Annealing

Progressively changes quantization profile

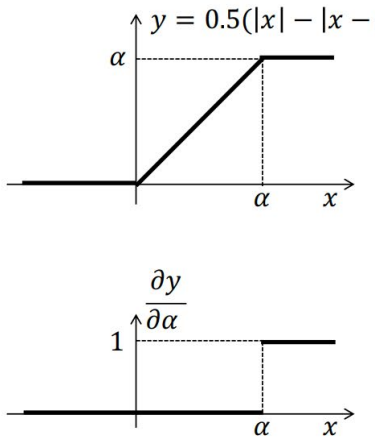
- (3,2)-(2,1)-(2,0)
- 3-bit/AR 2% at lr 0.1, 2-bit/AR 1% at lr 0.01, 2-bit/AR 0% at lr 0.001

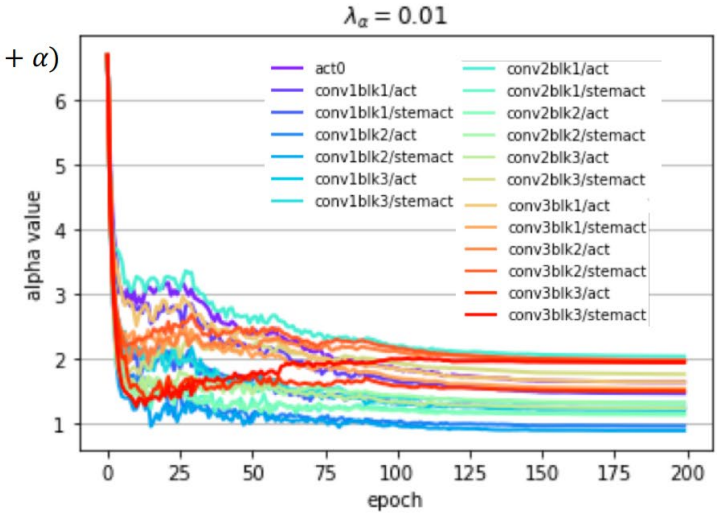


PArameterized Clipping acTivation (PACT) & Learned Step Size Quantization (LSQ)

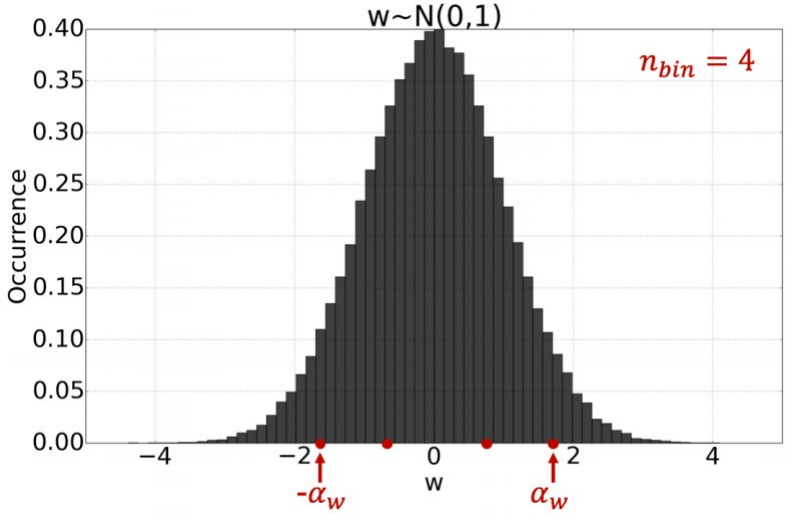
PArameterized Clipping acTivation (PACT)

- Limits the output range to $[0, \alpha]$
  - α is regularized by L2 regularization term
  - $y = PACT(x) = 0.5(|x| - |x - \alpha| + \alpha) = \begin{cases} 0, & x \in (-\infty, 0) \\ x, & x \in [0, \alpha) \\ \alpha, & x \in [\alpha, +\infty) \end{cases}$
- The truncated activation output is linearly quantized to k bits
  - $y_q = round\left(y \cdot \frac{2^k - 1}{\alpha}\right) \cdot \frac{\alpha}{2^k - 1}$
- STE-based approximation
  - $\frac{\partial y_q}{\partial \alpha} = \frac{\partial y_q}{\partial y}\frac{\partial y}{\partial \alpha} = \begin{cases} 0, x \in (-\infty, \alpha) \\ 1, x \in (\alpha, +\infty) \end{cases}$





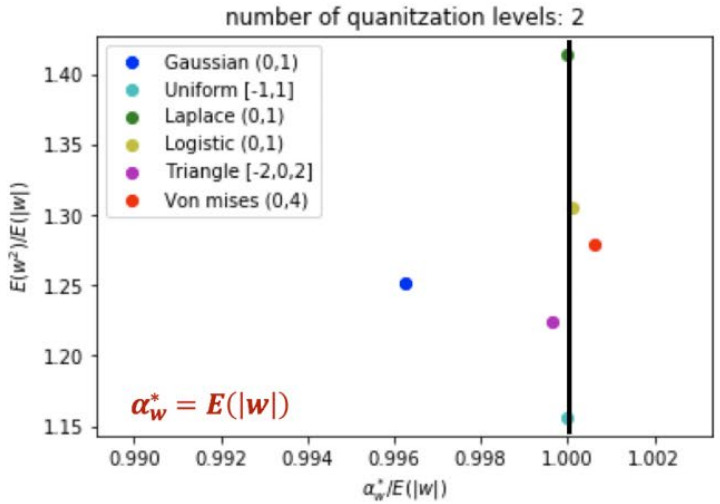
Statistics-Aware Weight Binning

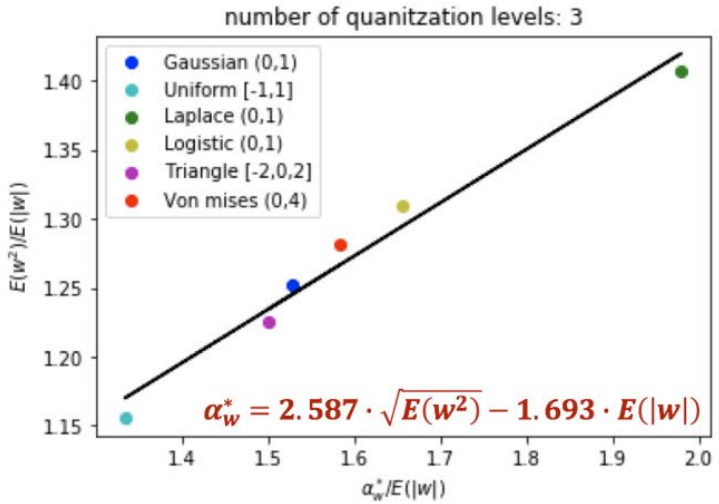


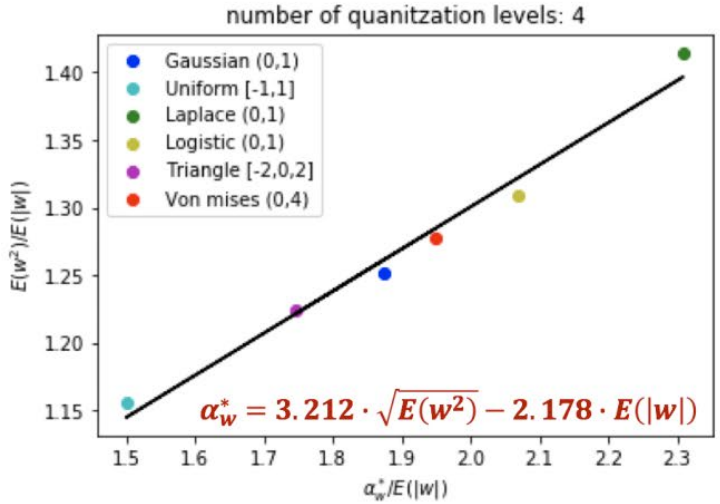
$$\alpha_w^* = \arg\min_{\alpha_w} ||w - w_q||^2$$

- Finding the optimal boundary is time-consuming
- The distribution of weight is different depending on the layer

- Numerically approximate the boundary based on $E(|w|)$ & $\sqrt{E(w^2)}$







Learned Step Size Quantization

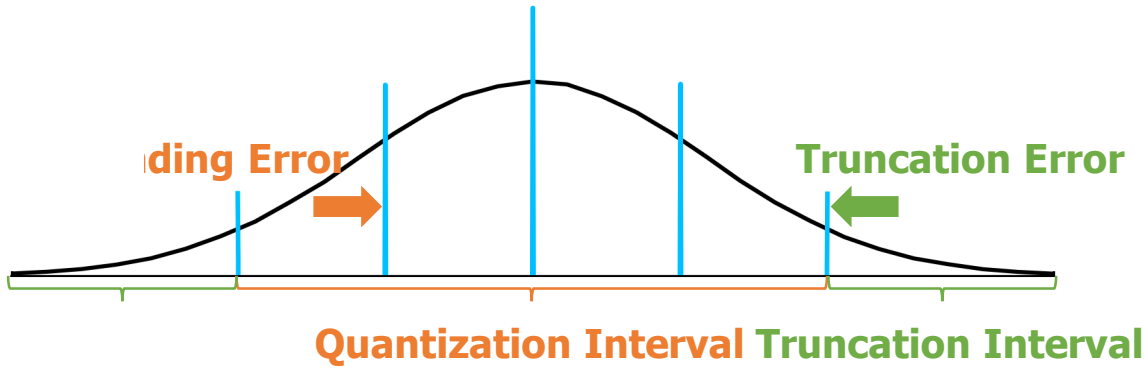
- There are two types of error induced by quantization
  - Rounding error
  - Truncation error

- Step size is trained to minimize both rounding and truncation errors

$$\bar{v} = \lfloor clip(v/s, -Q_N, Q_P) \rceil, \quad \hat{v} = \bar{v} \times s.$$

$$\frac{\partial \hat{v}}{\partial s} = \begin{cases} -v/s + \lfloor v/s \rceil & \text{if } -Q_N < v/s < Q_P \\ -Q_N & \text{if } v/s \leq -Q_N \\ Q_P & \text{if } v/s \geq Q_P \end{cases}$$

$$\frac{\partial \hat{v}}{\partial v} = \begin{cases} 1 & \text{if } -Q_N < v/s < Q_P \\ 0 & \text{otherwise,} \end{cases}$$



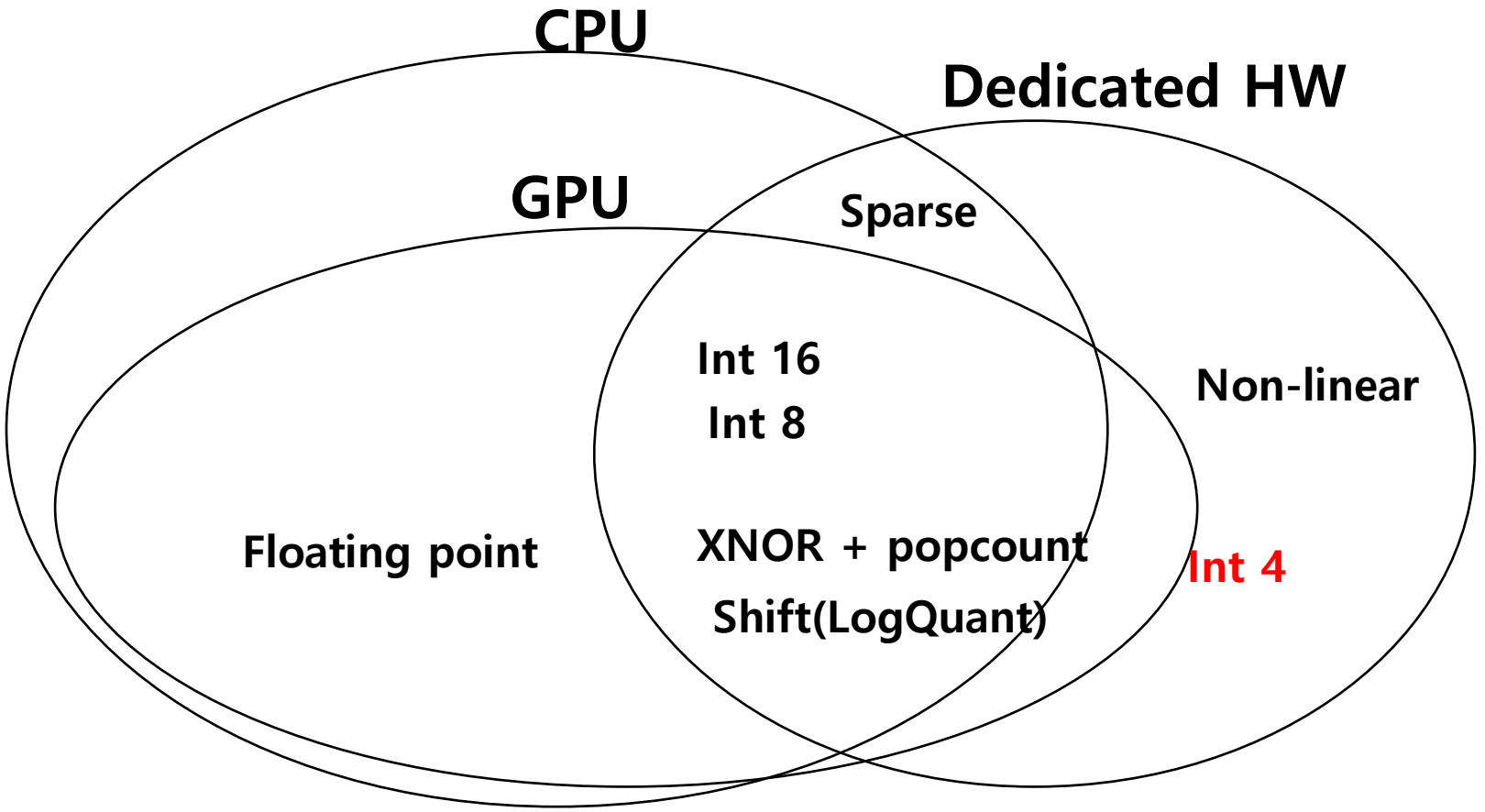
Results

Network	Method	Top-1 Accuracy @ Precision 2	3	4	8	Top-5 Accuracy @ Precision 2	3	4	8
ResNet-18		*Full precision: 70.5*				*Full precision: 89.6*			
	LSQ (Ours)	**67.6**	**70.2**	**71.1**	**71.1**	**87.6**	**89.4**	**90.0**	**90.1**
	QIL	65.7	69.2	70.1					
	FAQ			69.8	70.0			89.1	89.3
	LQ-Nets	64.9	68.2	69.3		85.9	87.9	88.8	
	PACT	64.4	68.1	69.2		85.6	88.2	89.0	
	NICE		67.7	69.8			87.9	89.21	
	Regularization	61.7		67.3	68.1	84.4		87.9	88.2
ResNet-34		*Full precision: 74.1*				*Full precision: 91.8*			
	LSQ (Ours)	**71.6**	**73.4**	**74.1**	**74.1**	**90.3**	**91.4**	**91.7**	**91.8**
	QIL	70.6	73.1	73.7					
	LQ-Nets	69.8	71.9			89.1	90.2		
	NICE		71.7	73.5			90.8	91.4	
	FAQ			73.3	73.7			91.3	91.6
ResNet-50		*Full precision: 76.9*				*Full precision: 93.4*			
	LSQ (Ours)	**73.7**	**75.8**	**76.7**	**76.8**	**91.5**	**92.7**	93.2	**93.4**
	PACT	72.2	75.3	76.5		90.5	92.6	93.2	
	NICE		75.1	76.5			92.3	**93.3**	
	FAQ			76.3	76.5			92.9	93.1
	LQ-Nets	71.5	74.2	75.1		90.3	91.6	92.4	
ResNet-101		*Full precision: 78.2*				*Full precision: 94.1*			
	LSQ (Ours)	**76.1**	**77.5**	**78.3**	**78.1**	**92.8**	**93.6**	**94.0**	**94.0**
ResNet-152		*Full precision: 78.9*				*Full precision: 94.3*			
	LSQ (Ours)	**76.9**	**78.2**	**78.5**	**78.5**	**93.2**	**93.9**	**94.1**	**94.2**
	FAQ			78.4	**78.5**			**94.1**	94.1
VGG-16bn		*Full precision: 73.4*				*Full precision: 91.5*			
	LSQ (Ours)	**71.4**	**73.4**	**74.0**	73.5	**90.4**	**91.5**	**92.0**	**91.6**
	FAQ			73.9	**73.7**			91.7	**91.6**
Squeeze Next-23-2x		*Full precision: 67.3*				*Full precision: 87.8*			
	LSQ (Ours)	**53.3**	**63.7**	**67.4**	**67.0**	**77.5**	**85.4**	**87.8**	**87.7**

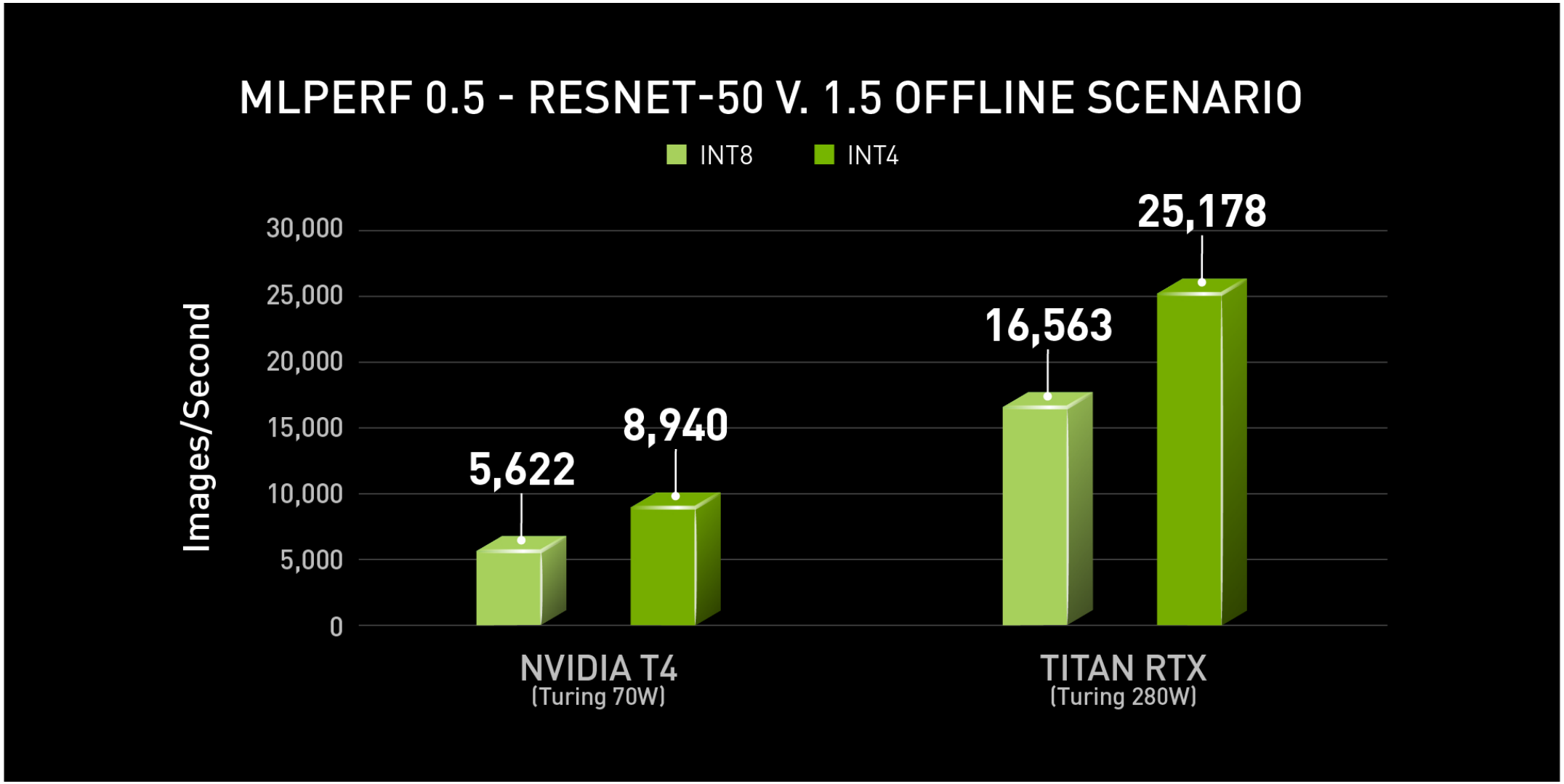
PROFIT: A Novel Training Method for sub-4-bit MobileNet Models

HW-Friendly Computation

- Quantization is the task of restricting the data representation
  - Quantization could be beneficial to computation



4-bit Quantization?



https://devblogs.nvidia.com/int4-for-ai-inference/

Quantization Trend

Conventional Networks

AlexNet
Top-1: 61.8 %
727 MFLOPs

2016	2017	2018	2019
4-bit, 54.9 %	-	4-bit, 55.7 %	4-bit, 62.0 %
2-bit, 53.6 %	2-bit, 55.7 %	2-bit, 55.0 %	2-bit, 58.1 %
DoReFa-Net	Balanced-Q	PACT	QIL

ResNet-18
Top-1: 70.5 %
2 GFLOPs

2016	2018	2019	2020
4-bit, 68.1 %	4-bit, 69.2 %	4-bit, 70.1 %	4-bit, 71.1 %
2-bit, 62.6 %	2-bit, 64.4 %	2-bit, 65.7 %	2-bit, 67.6 %
DoReFa-Net	PACT	QIL	LSQ

Optimized Networks

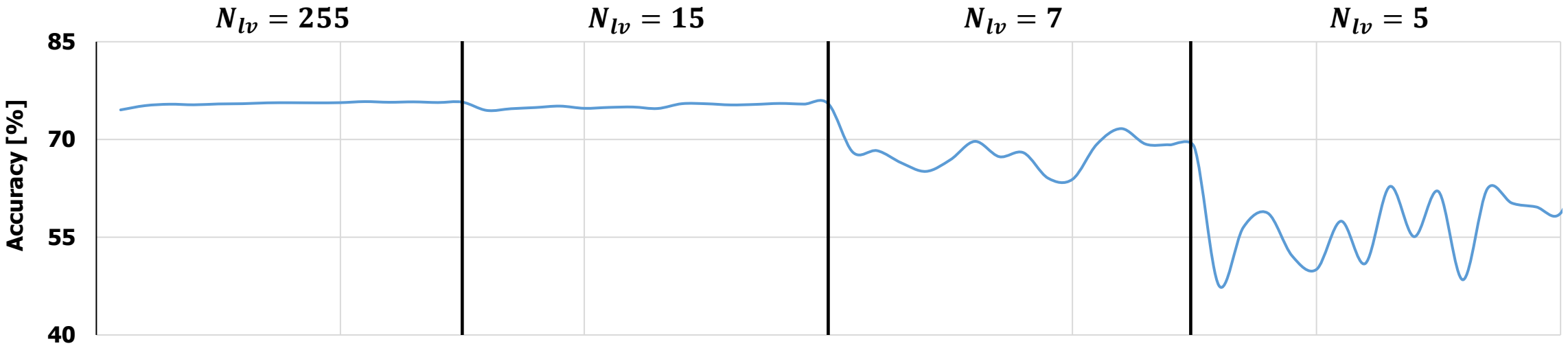
MobileNet-v2
Top-1: 71.9 %
300 MFlops

2018	2019	2020
4-bit, 61.4 %	4-bit, 64.8 %	**4-bit, 71.5 %**
PACT	DSQ	**Ours**

MobileNet-v3
Top-1: 75.3 %
217 MFlops

2018	2019	2020
4-bit, 57.09 %	4-bit, fail	**4-bit, 73.8 %**
PACT	QIL	**Ours**

Activation Instability



Accuracy curve of MobileNet-v3 with weight quantization for CIFAR100 dataset

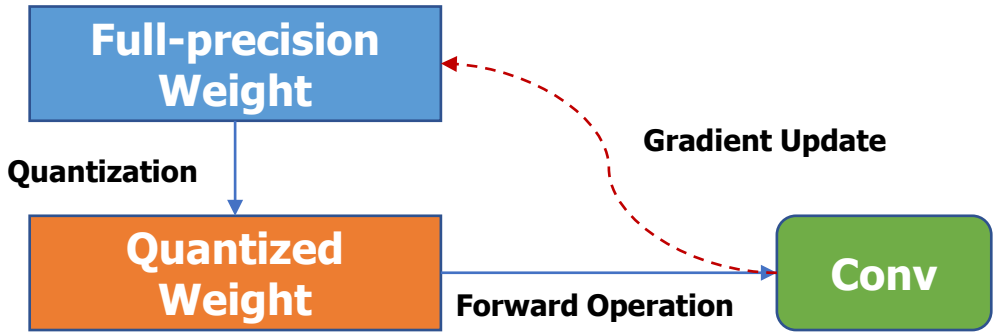
- The test accuracy becomes unstable when the weight bit-width is reduced

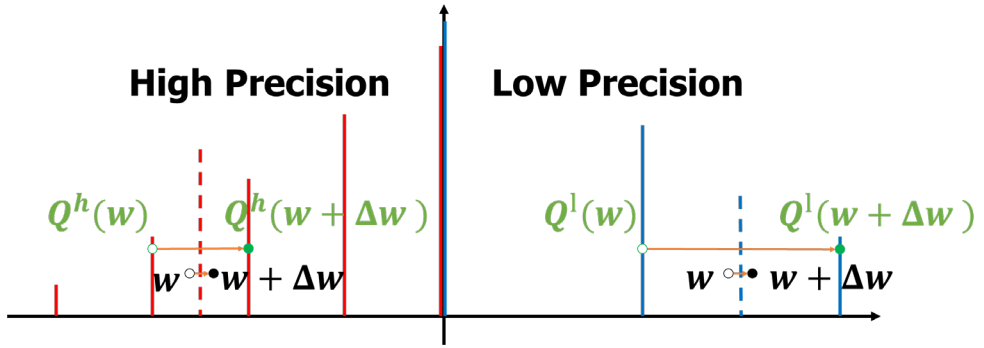
Activation Instability

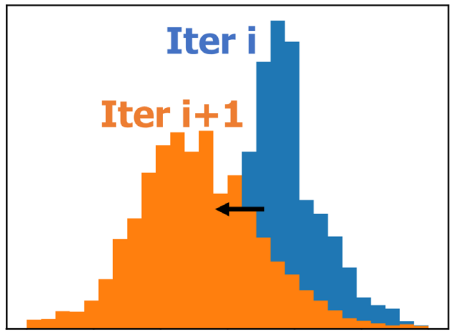
Fine-tuning is essential to recover the accuracy drop

Weight is updated during fine-tuning

- STE based quantized weight update





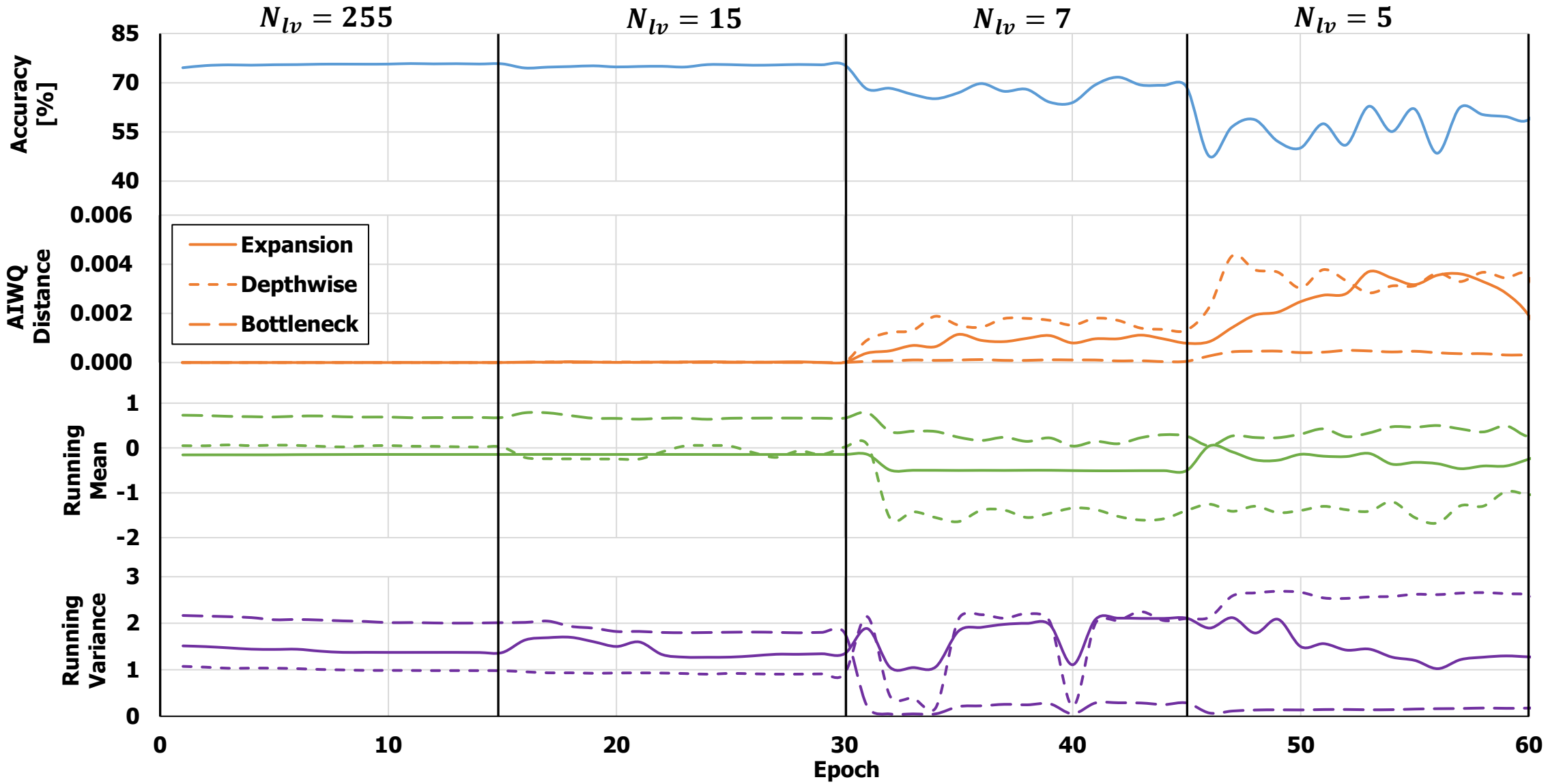


Metric definition for activation instability based on KL-divergence

$$p_o^t \approx N\left(\mu_o, \sigma_o \,||\, \sum_i Q(W_{l,o,i}^t) \otimes I_{l,i}^t\right) \qquad q_o^t \approx N\left(\mu_o', \sigma_o' \,||\, \sum_i Q(W_{l,o,i}^{t-1}) \otimes I_{l,i}^t\right)$$

$$D^l = E_o^t\left[D_{KL}(p_o^t \,||\, q_o^t)\right]$$

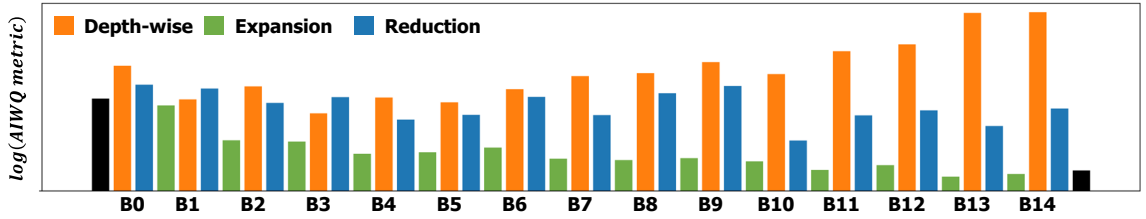
AIWQ Problem and Accuracy

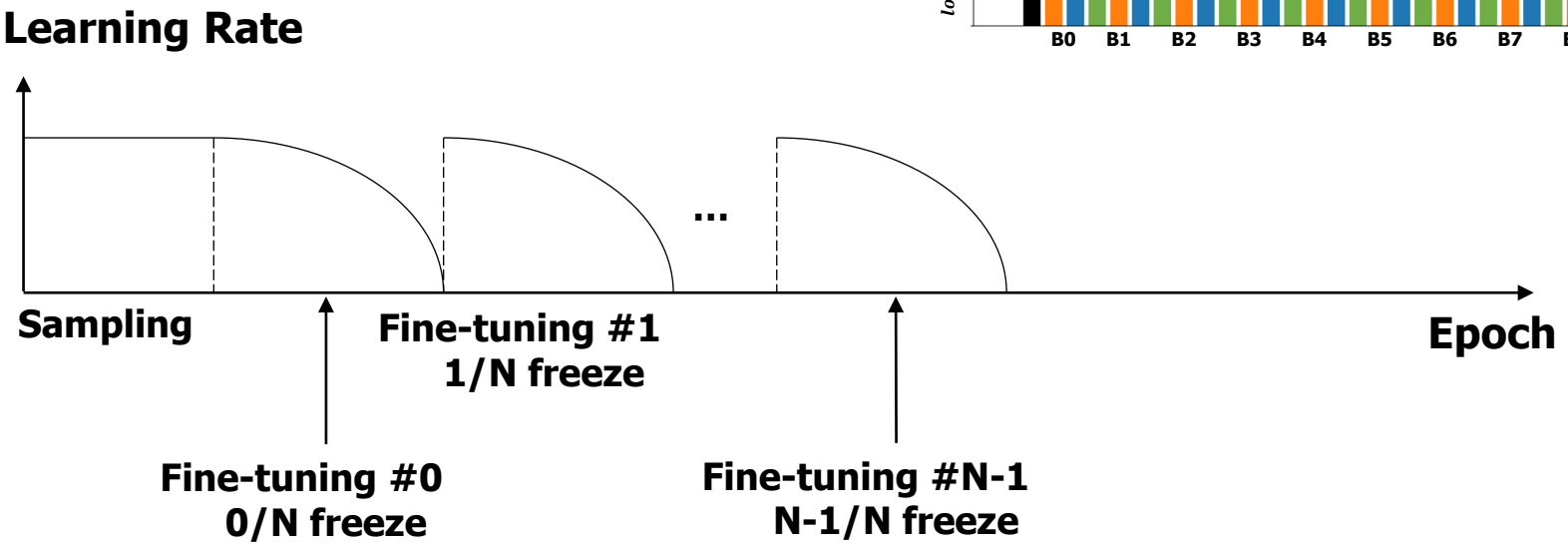


PROFIT

PROgressively Freezing Iterative Training (PROFIT)

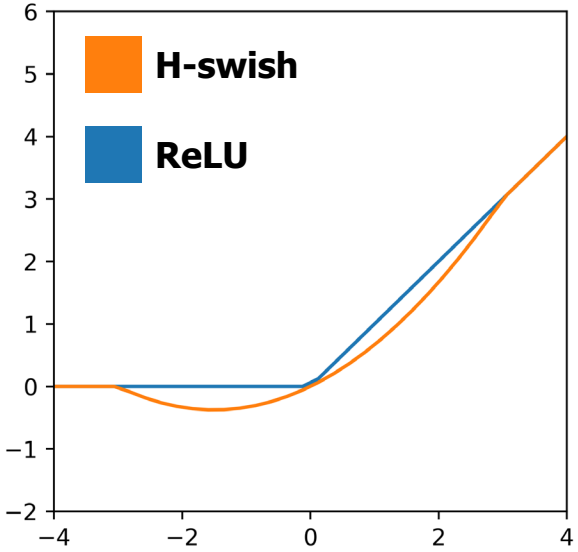
1. Layer-wise AIWQ sensitivity analysis based on AIWQ metric
2. Perform fine-tuning
3. Select top-k sensitive layers among the trained layers and set the learning rate to 0
4. Perform fine-tuning for the rest layers
5. Repeat 3 to 4 until all layers are frozen



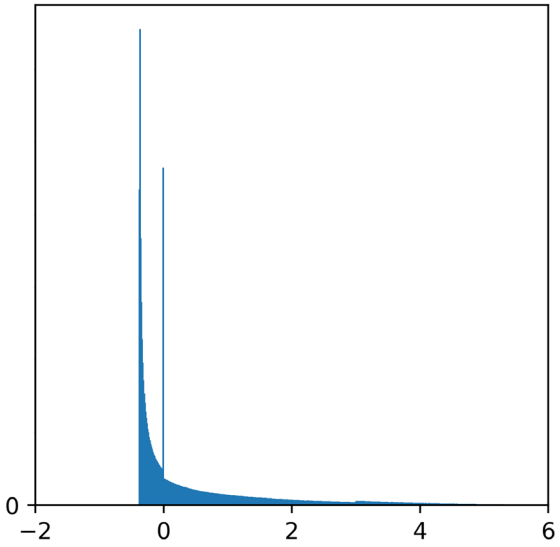


Asymmetric Distribution

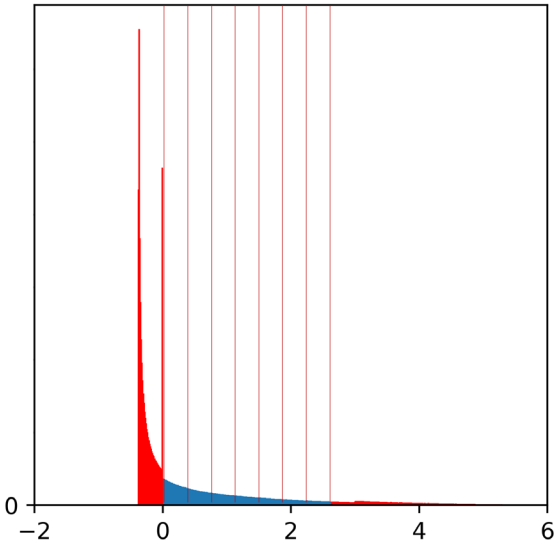
- Existing quantization algorithms only support non-negative or symmetric data
  - Small negative values are ignored
  - Quantization levels are wasted, and a truncation error is occurred



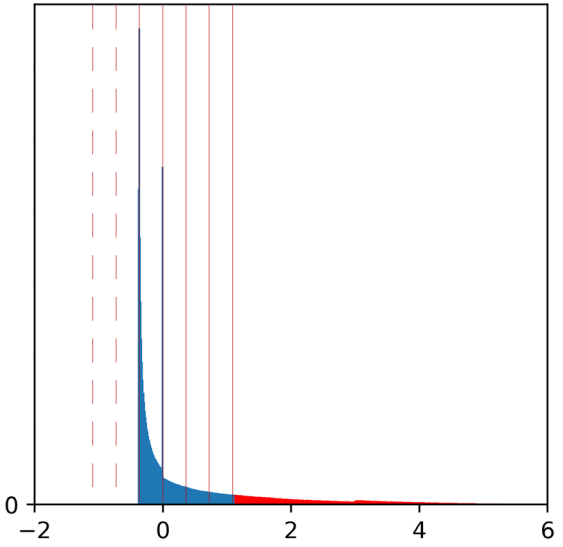
(a) Asymmetric range of h-swish function



(b) Output histogram of h-swish function

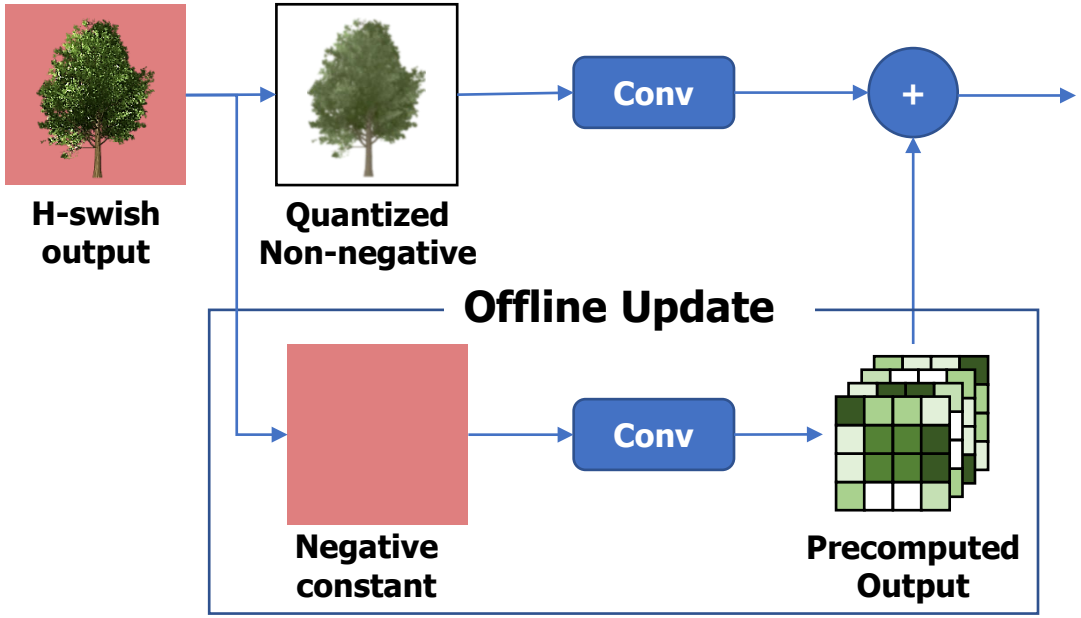


(b) Non-negative quantization

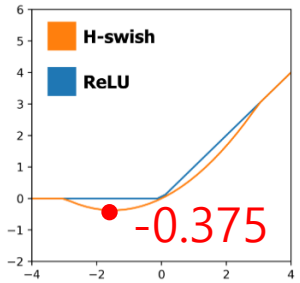


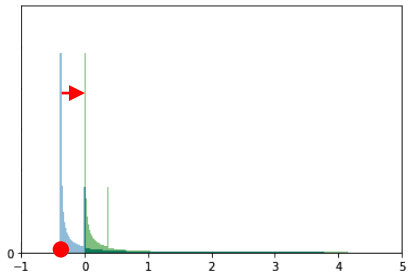
(b) Symmetric quantization

Negative Padding

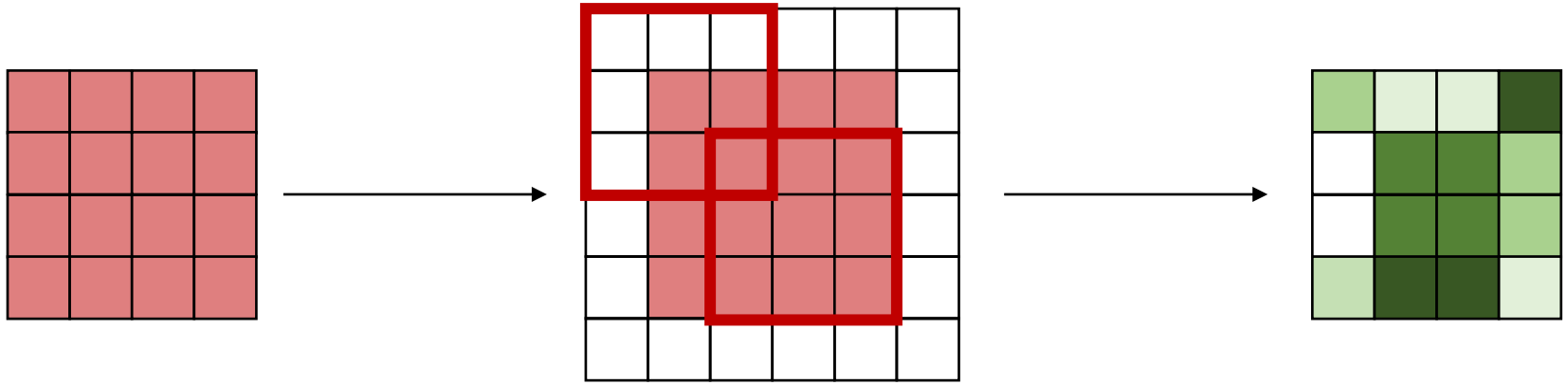


- Solution candidates

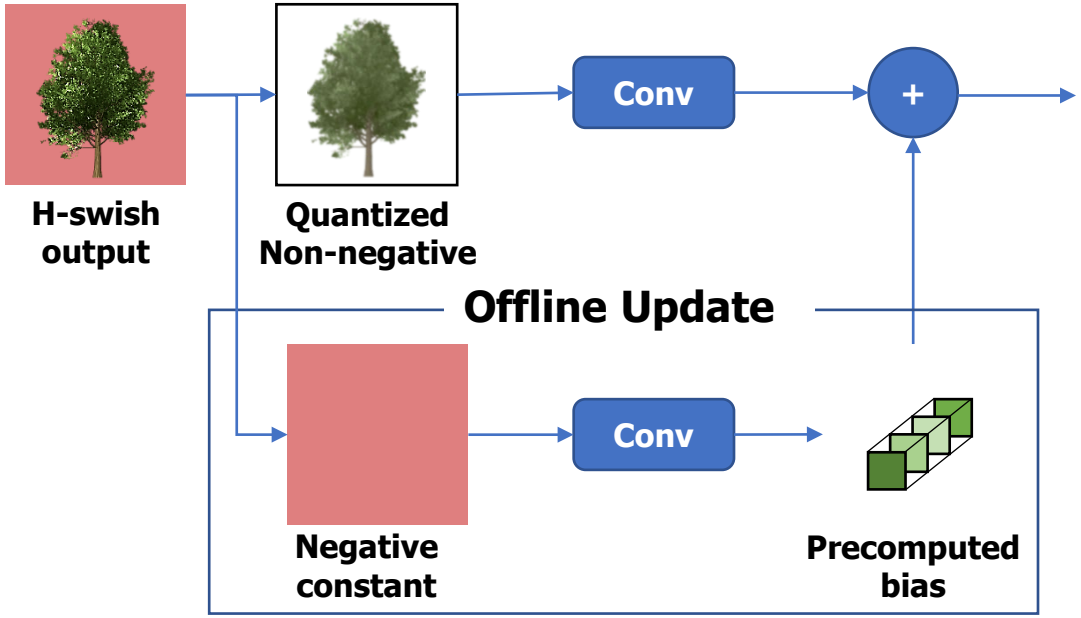




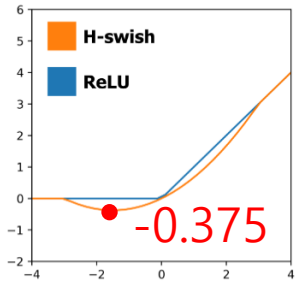
- When using traditional zero padding,
  the output values of the center and edge are different
  - Ex) 3x3 conv with zero padding

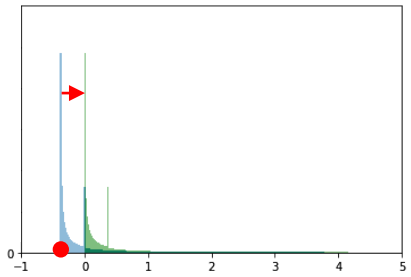


Negative Padding

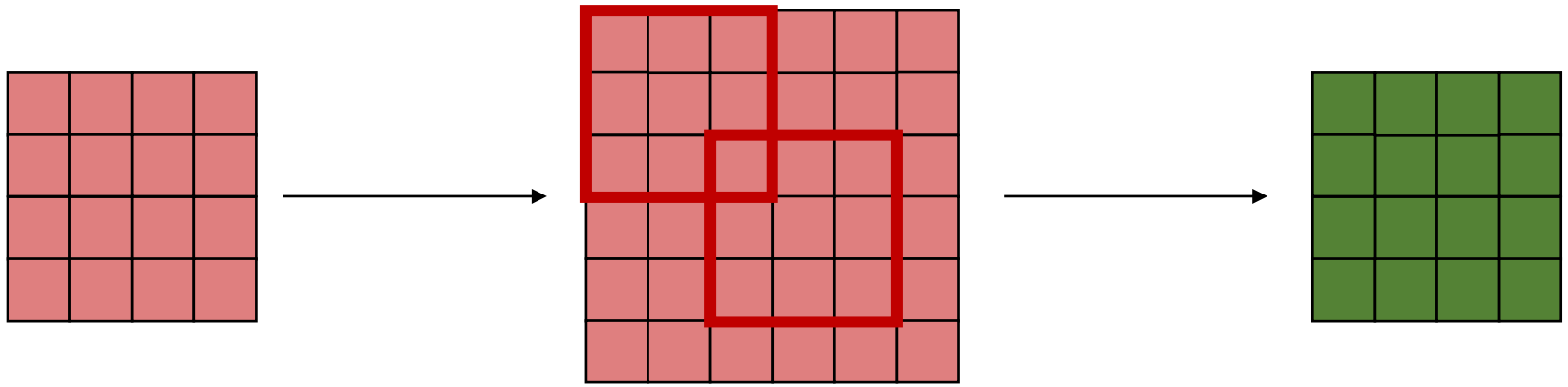


- Solution candidates





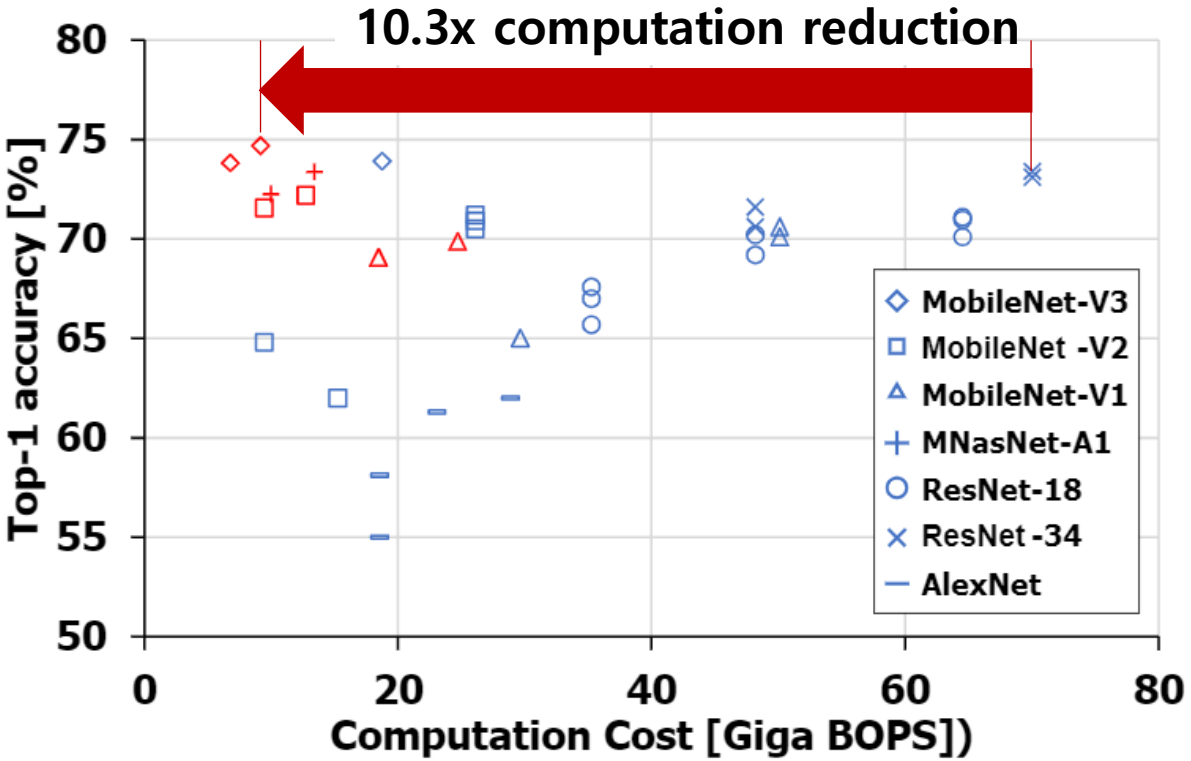
- Instead of using zero padding, use negative padding (minimum-value)
  - Ex) 3x3 conv with -0.375 padding

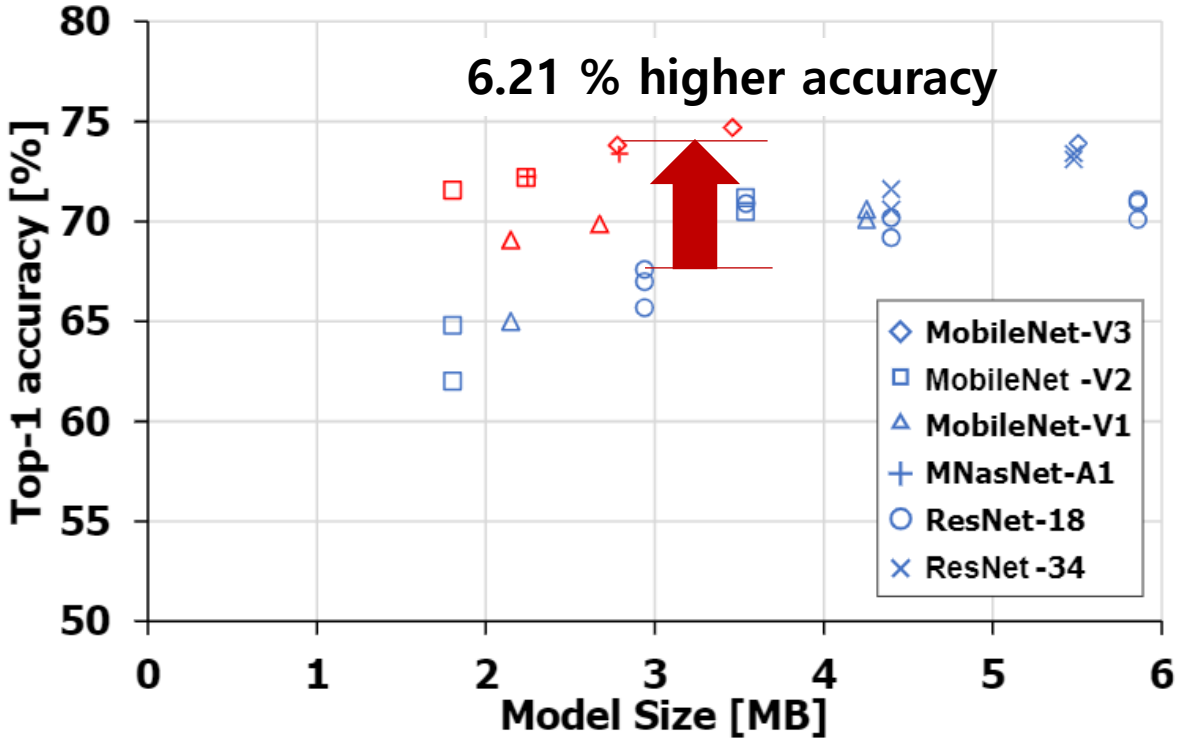


Result

	MobileNet-v1	MobileNet-v2	MobileNet-v3	MNasNet-A1
Full	68.848 / 88.740	71.328 / 90.016	74.728 / 92.136	73.130 / 91.276
Full+	69.552 / 89.138	71.944 / 90.470	75.296 / 92.446	73.396 / 91.464
8-bit	70.164 / 89.370	72.352 / 90.636	75.166 / 92.498	73.742 / 91.756
5-bit	69.866 / 89.058	72.192 / 90.498	74.690 / 92.092	73.378 / 91.244
4-bit	69.056 / 88.412	71.564 / 90.398	73.812 / 91.588	72.244 / 90.584
	-0.496 %	-0.380 %	-1.484 %	

Result


10.3x computation reduction
Top-1 accuracy [%]
80
75
70
65
60
55
50
0
20
40
60
80
Computation Cost [Giga BOPS])
MobileNet-V3
MobileNet -V2
MobileNet-V1
MNasNet-A1
ResNet-18
ResNet -34
AlexNet


6.21 % higher accuracy
Top-1 accuracy [%]
80
75
70
65
60
55
50
0
1
2
3
4
5
6
Model Size [MB]
MobileNet-V3
MobileNet -V2
MobileNet-V1
MNasNet-A1
ResNet-18
ResNet -34