Renmin University of China
North Carolina State University

Space Efficient TREC for Enabling Deep Learning on Microcontrollers

Jiawei Guan⋄, Feng Zhang⋄, Jiesong Liu⋄, Hsin-Hsuan Sung⋆, Ruofan Wu⋄, Xiaoyong Du⋄, Xipeng Shen⋆

⋄Key Laboratory of Data Engineering and Knowledge Engineering (MOE), and School of Information, Renmin University of China
⋆Computer Science Department, North Carolina State University

Outline

- Motivation
- TREC Architecture
- Experiments
- Results

Motivation

AIoT

- CNN Inference on the edge.

Motivation

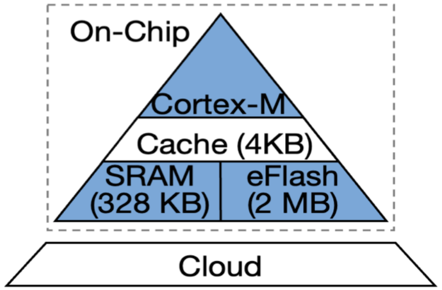
AIoT

- CNN Inference on the edge.
- Computation-intensive.

Motivation

AIoT

- CNN Inference on the edge.
- Computation-intensive.
- Resource-constrained devices.

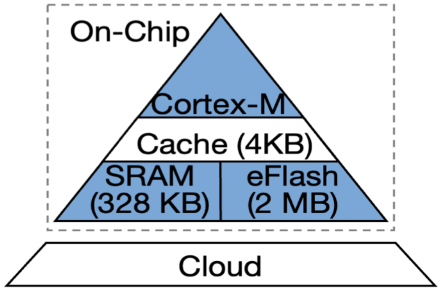


Memory Hierarchy for Cortex-M4.

Motivation

AIoT

- CNN Inference on the edge.
- Computation-intensive.
- Resource-constrained devices.
- Large models are required to be compressed.

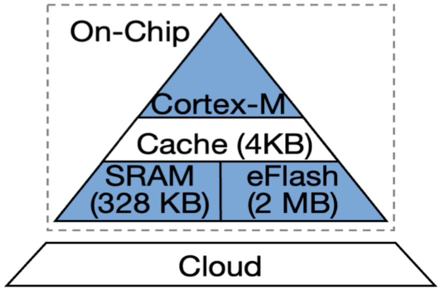


Memory Hierarchy for Cortex-M4.

Motivation

AIoT

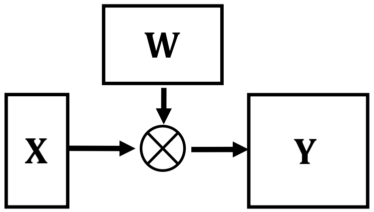
- CNN Inference on the edge.
- Computation-intensive.
- Resource-constrained devices.
- Large models are required to be compressed.



Memory Hierarchy for Cortex-M4.

Redundancy

- Lasting redundancy
- Transient redundancy

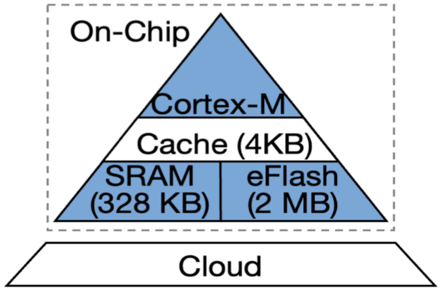


Convolutional Layer

Motivation

AIoT

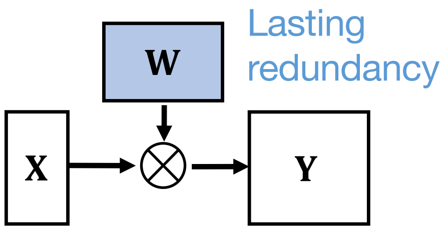
- CNN Inference on the edge.
- Computation-intensive.
- Resource-constrained devices.
- Large models are required to be compressed.



Memory Hierarchy for Cortex-M4.

Redundancy

- Lasting redundancy
  - Arises from DNN parameters.
  - Removed during training mostly.
  - Pruning, quantization, etc.
- Transient redundancy

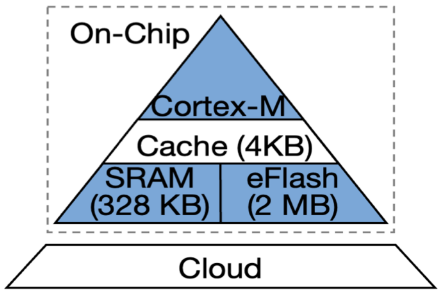


Convolutional Layer

Motivation

AIoT

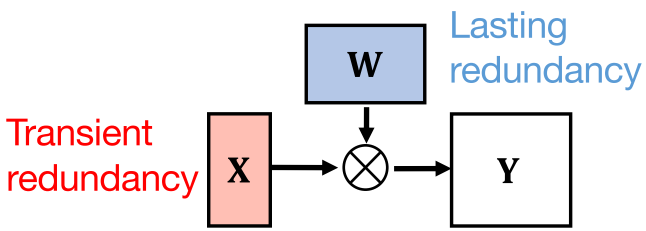
- CNN Inference on the edge.
- Computation-intensive.
- Resource-constrained devices.
- Large models are required to be compressed.



Memory Hierarchy for Cortex-M4.

Redundancy

- Lasting redundancy
  - Arises from DNN parameters.
  - Removed during training mostly.
  - Pruning, quantization, etc.
- Transient redundancy
  - Arises from Input data / activation maps.
  - Removed during inference.

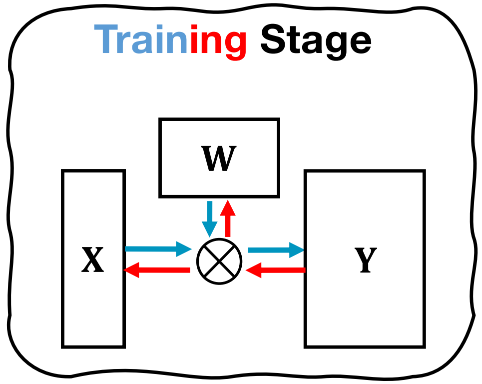


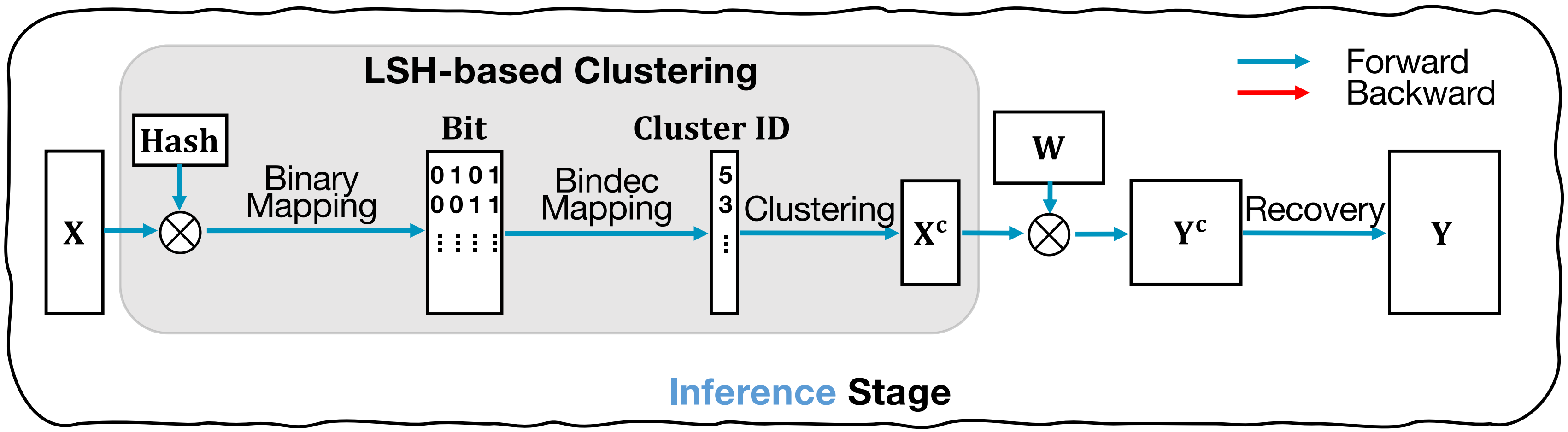
Convolutional Layer

Motivation

DeepReuse

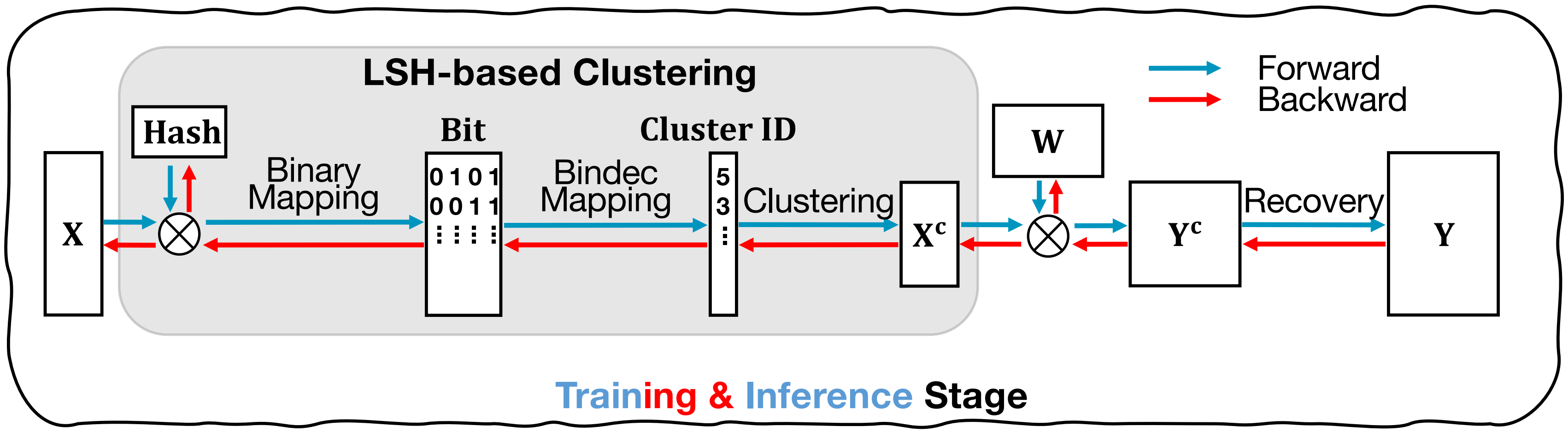
- Treat transient redundancy in an Ad-hoc manner.
- Over 5% accuracy fluctuations in different runs.



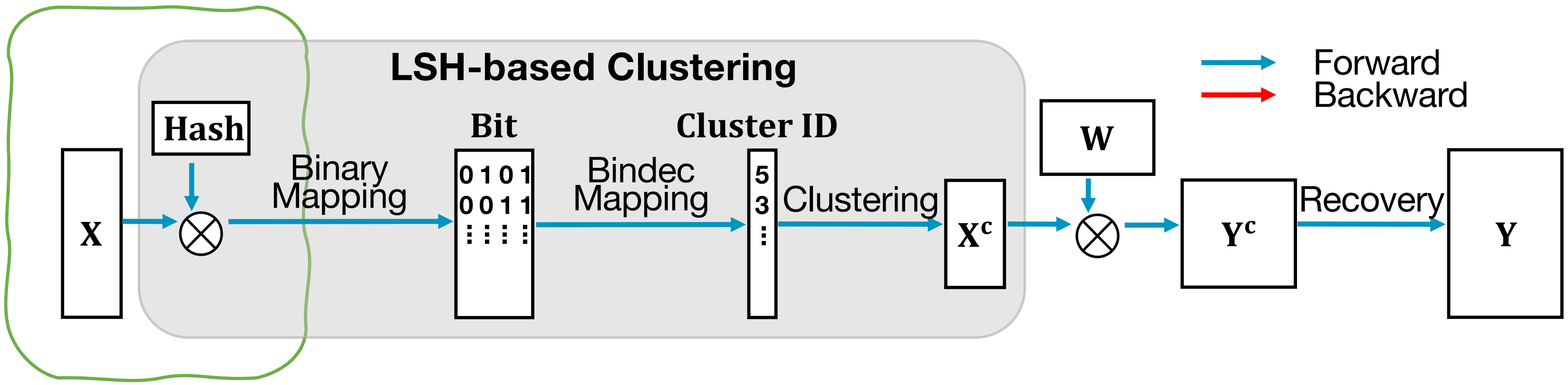


Concept

Can the detection and elimination of transient redundancy be integrated into the CNN training process, so that they become part of its inherent architecture?



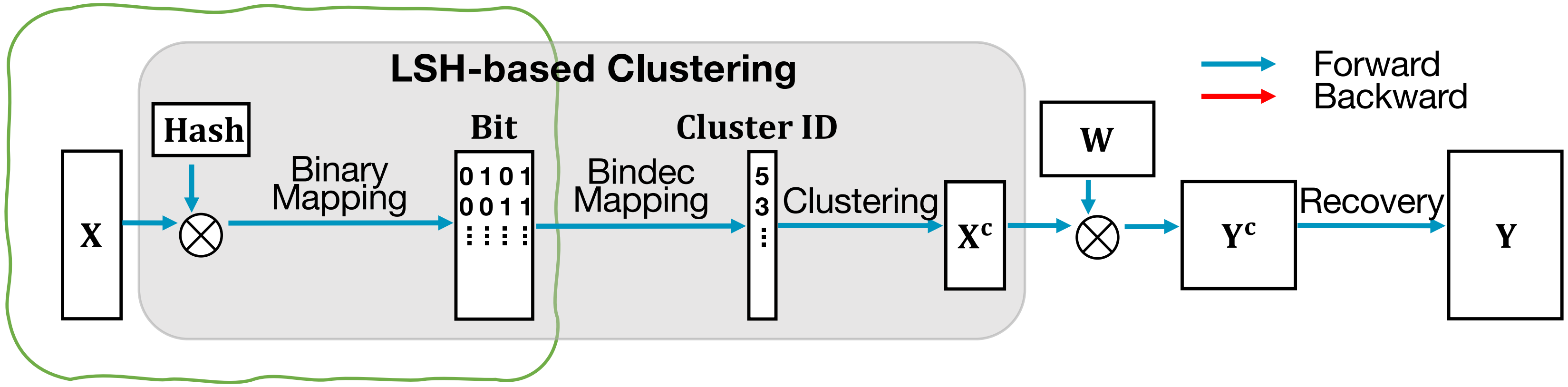
A Closer Look at Redundancy Elimination



$$\overset{\boldsymbol{X}}{\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 3 & 2 \end{bmatrix}} \times \overset{\text{Hash}}{\begin{bmatrix} -1 & 1 \\ 1 & 2 \\ 0 & -1 \end{bmatrix}} = \overset{\text{Projected}}{\begin{bmatrix} 1 & 2 \\ 0 & -1 \\ 2 & 3 \\ 2 & 5 \end{bmatrix}}$$

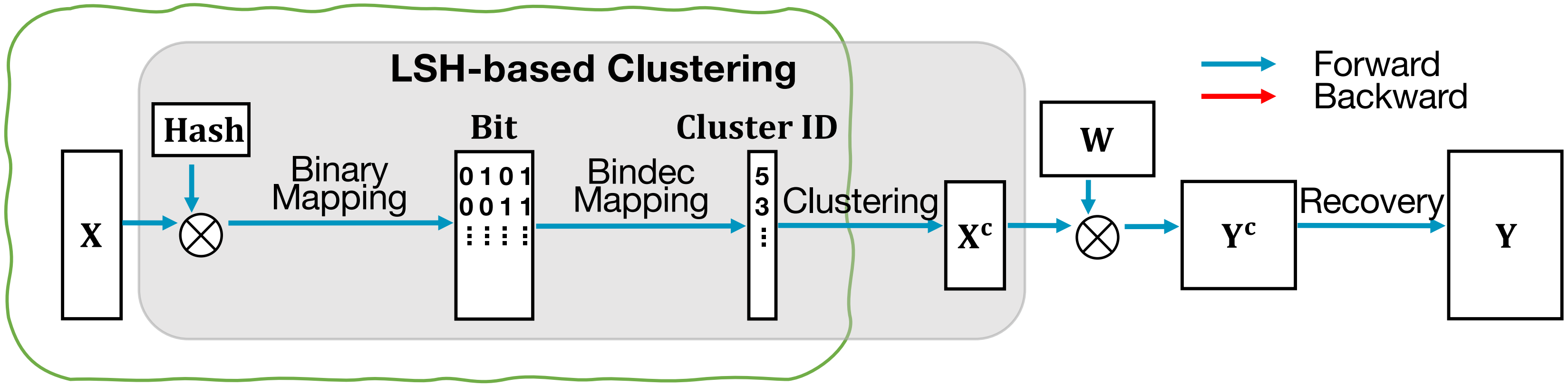
Renmin University of China
North Carolina State University

A Closer Look at Redundancy Elimination



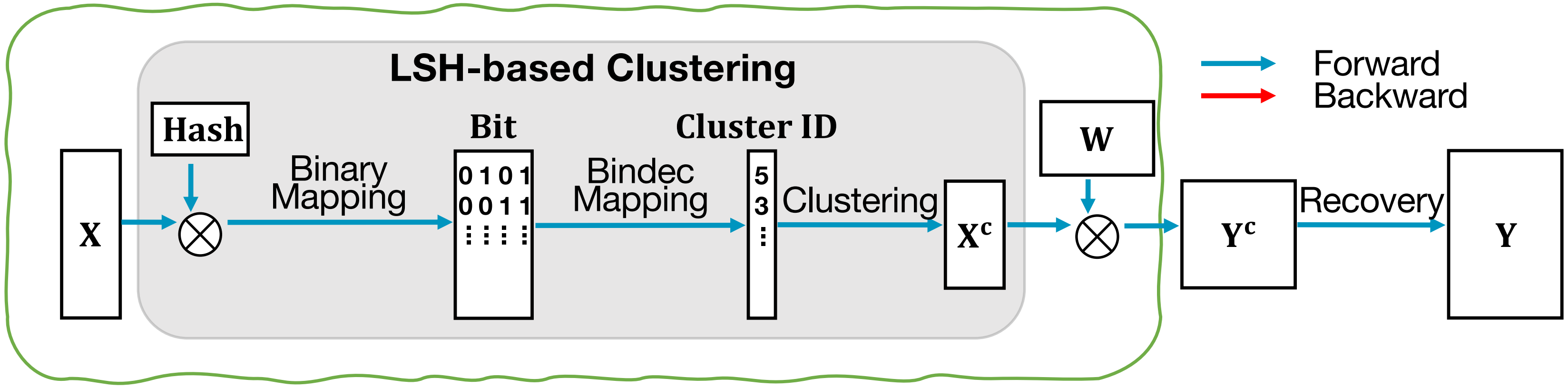
$$\overset{\boldsymbol{X}}{\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 3 & 2 \end{bmatrix}} \times \overset{\text{Hash}}{\begin{bmatrix} -1 & 1 \\ 1 & 2 \\ 0 & -1 \end{bmatrix}} = \overset{\text{Projected}}{\begin{bmatrix} 1 & 2 \\ 0 & -1 \\ 2 & 3 \\ 2 & 5 \end{bmatrix}} \xrightarrow[\text{Mapping}]{\text{Binary}} \overset{\text{Bit}}{\begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}}$$

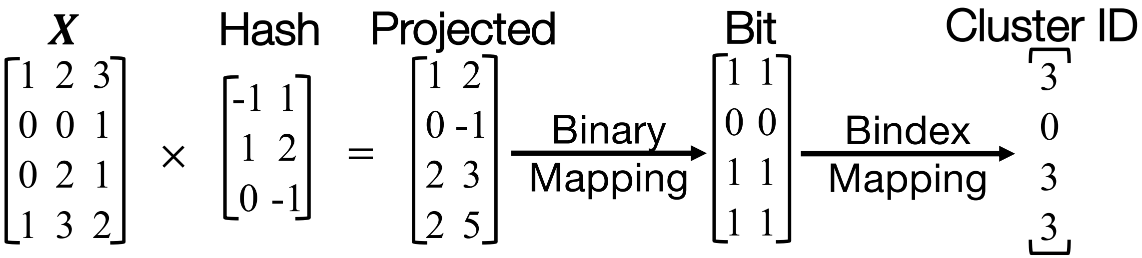
A Closer Look at Redundancy Elimination



$$\underset{X}{\begin{bmatrix}1 & 2 & 3\\ 0 & 0 & 1\\ 0 & 2 & 1\\ 1 & 3 & 2\end{bmatrix}} \times \underset{\text{Hash}}{\begin{bmatrix}-1 & 1\\ 1 & 2\\ 0 & -1\end{bmatrix}} = \underset{\text{Projected}}{\begin{bmatrix}1 & 2\\ 0 & -1\\ 2 & 3\\ 2 & 5\end{bmatrix}} \xrightarrow[\text{Mapping}]{\text{Binary}} \underset{\text{Bit}}{\begin{bmatrix}1 & 1\\ 0 & 0\\ 1 & 1\\ 1 & 1\end{bmatrix}} \xrightarrow[\text{Mapping}]{\text{Bindex}} \underset{\text{Cluster ID}}{\begin{bmatrix}3\\ 0\\ 3\\ 3\end{bmatrix}}$$

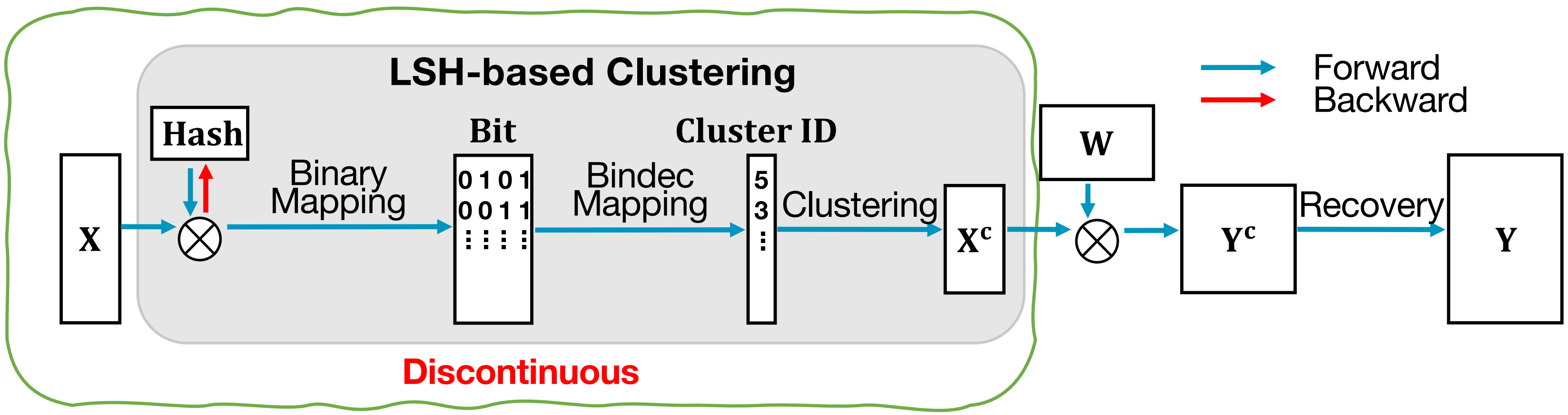
A Closer Look at Redundancy Elimination



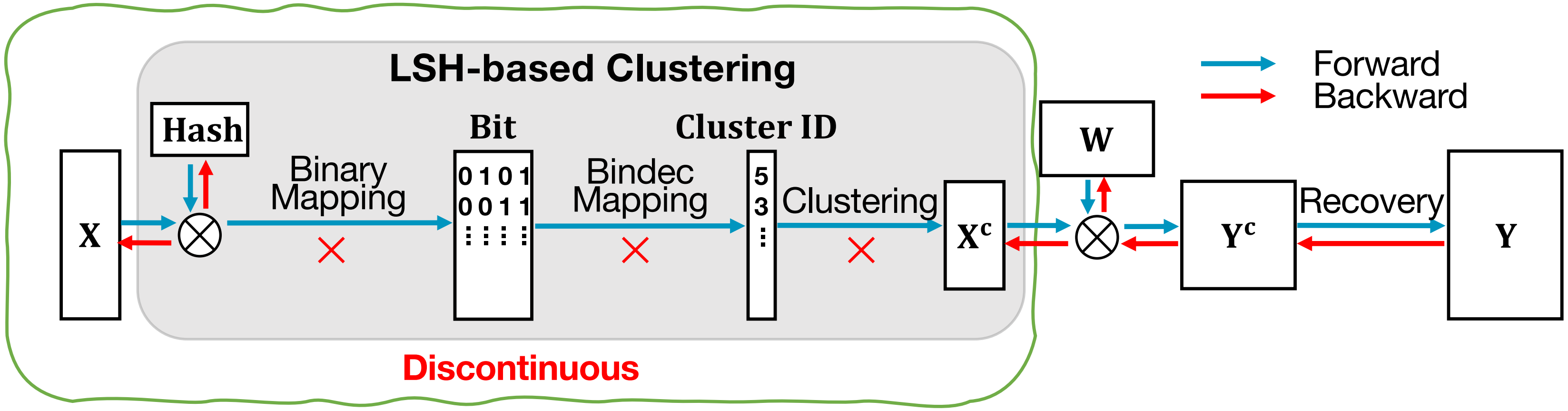


$$\underset{X}{\begin{bmatrix}1&2&3\\0&0&1\\0&2&1\\1&3&2\end{bmatrix}} \times \underset{\text{Hash}}{\begin{bmatrix}-1&1\\1&2\\0&-1\end{bmatrix}} = \underset{\text{Projected}}{\begin{bmatrix}1&2\\0&-1\\2&3\\2&5\end{bmatrix}} \xrightarrow[\text{Mapping}]{\text{Binary}} \underset{\text{Bit}}{\begin{bmatrix}1&1\\0&0\\1&1\\1&1\end{bmatrix}} \xrightarrow[\text{Mapping}]{\text{Bindex}} \underset{\text{Cluster ID}}{\begin{bmatrix}3\\0\\3\\3\end{bmatrix}}$$

Backward Pass

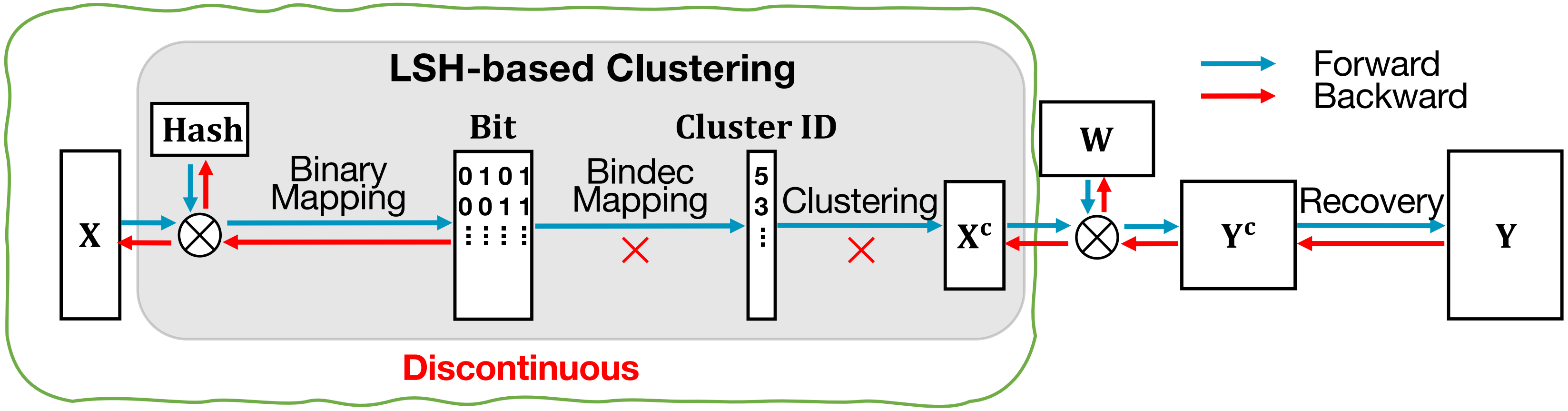


Backward Pass



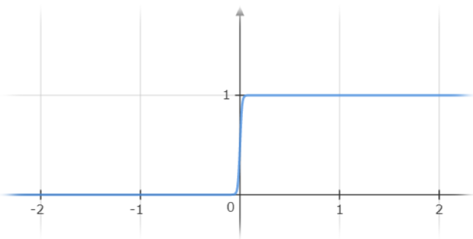
Renmin University of China
North Carolina State University

Backward Pass



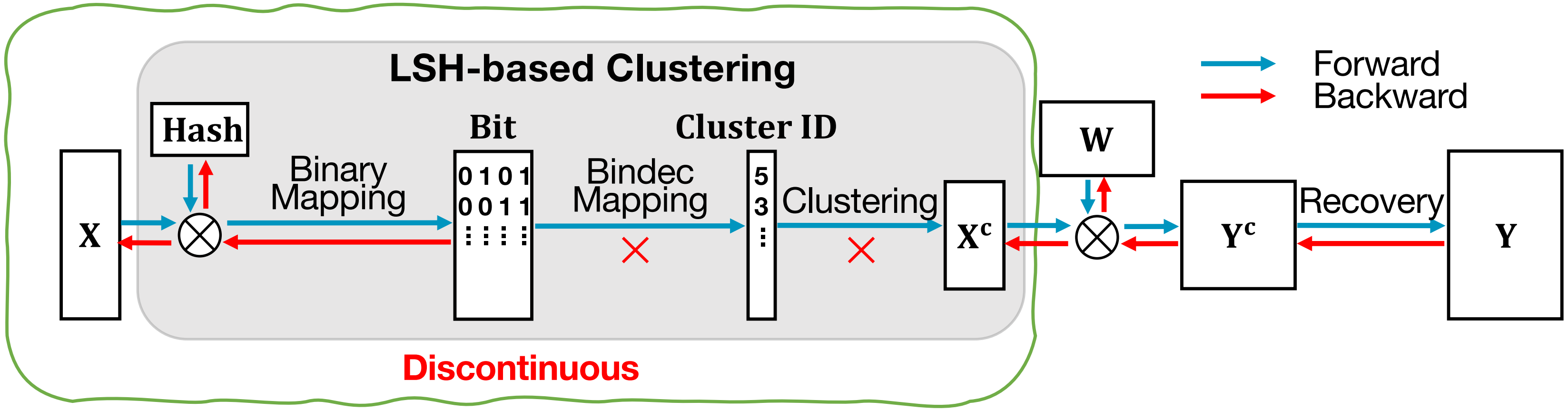
1. Binary Mapping → Binary Approximation

$$sigmoid(x) = \frac{1}{1 + e^{-\alpha x}}$$



Renmin University of China
North Carolina State University

Backward Pass

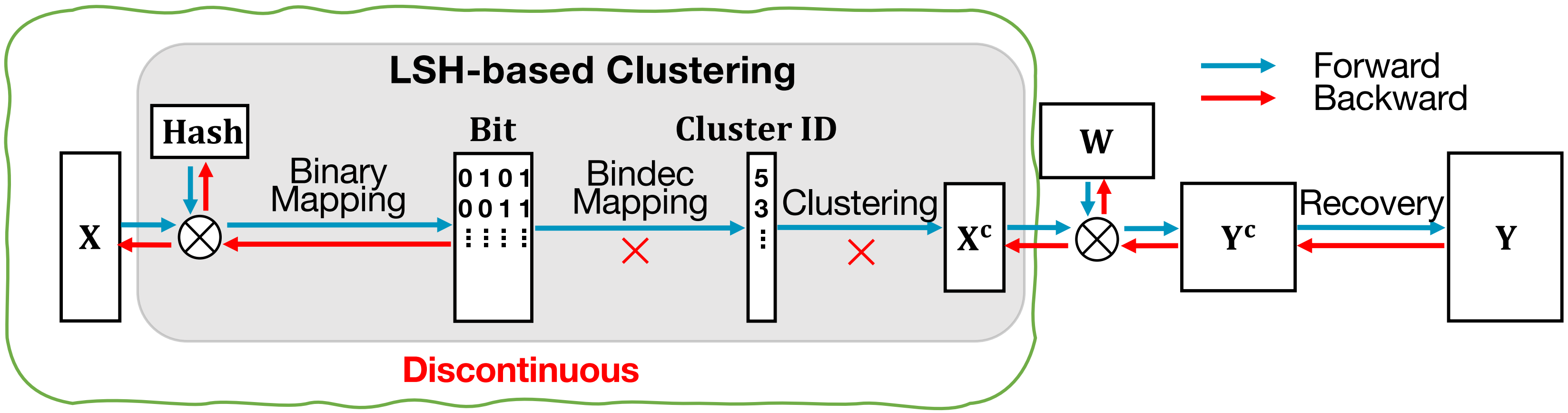


2. Bindec Mapping → Bindec Conversion

- Transformation matrix

$$\begin{bmatrix} 2^{H-1}/1 & 2^{H-1}/2 & \cdots & 2^{H-1}/2^H \\ \vdots & \vdots & \ddots & \vdots \\ 2^1/1 & 2^1/2 & \cdots & 2^1/2^H \\ 2^0/1 & 2^0/2 & \cdots & 2^0/2^H \end{bmatrix}$$

Backward Pass



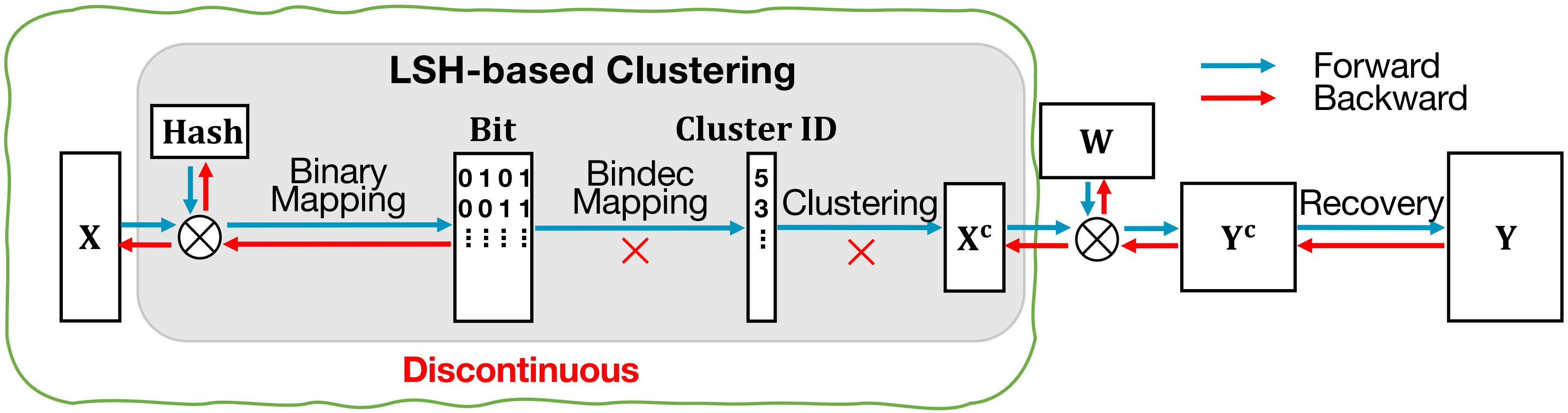
Bindec Conversion

Bit — Transformation ($H\times2^H$) — Quotient

$$\left(\begin{bmatrix}1 & 1\\0 & 0\\1 & 1\\1 & 1\end{bmatrix}+\begin{bmatrix}0 & 1\\0 & 1\\0 & 1\\0 & 1\end{bmatrix}\right)\times\begin{bmatrix}2^1/1 & 2^1/2 & 2^1/3 & 2^1/4\\2^0/1 & 2^0/2 & 2^0/3 & 2^0/4\end{bmatrix}=\begin{bmatrix}4 & 2 & 4/3 & 1\\1 & 1/2 & 1/3 & 1/4\\4 & 2 & 4/3 & 1\\4 & 2 & 4/3 & 1\end{bmatrix}$$

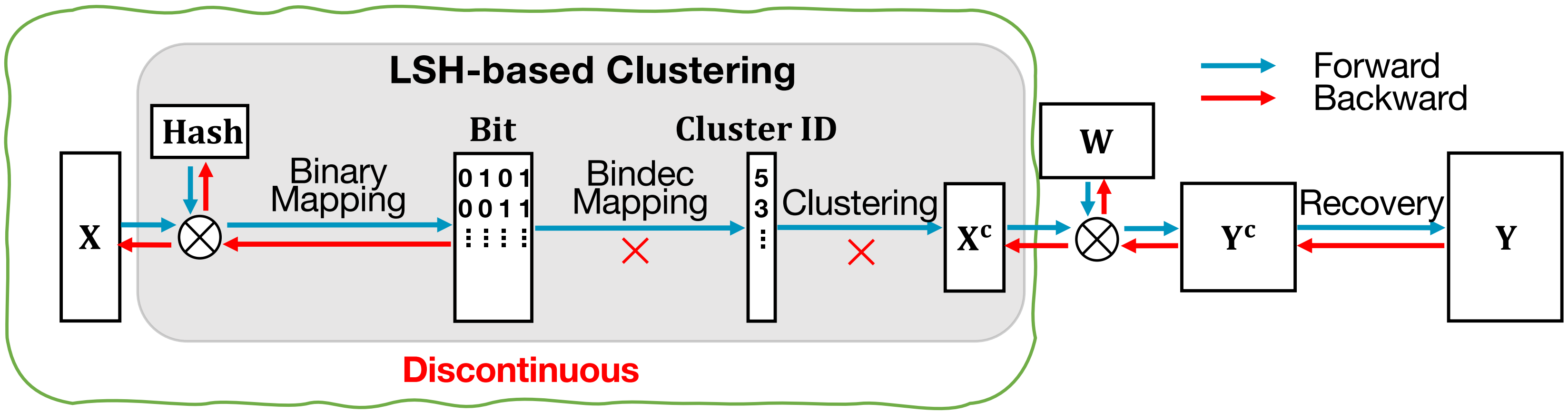
Renmin University of China
North Carolina State University

Backward Pass



$$\begin{array}{cccc} \textbf{Bit} & & \textbf{Transformation } (\mathbf{H\times 2^{H}}) & & \textbf{Quotient} \\ \left(\begin{bmatrix}1&1\\0&0\\1&1\\1&1\end{bmatrix}+\begin{bmatrix}0&1\\0&1\\0&1\\0&1\end{bmatrix}\right) & \times & \begin{bmatrix}2^1/&2^1/&2^1/&2^1/\\2^0/&2^0/&2^0/&2^0/\end{bmatrix} & = & \begin{bmatrix}4&4&4&4\\1&1&1&1\\4&4&4&4\\4&4&4&4\end{bmatrix}\end{array}$$

Backward Pass



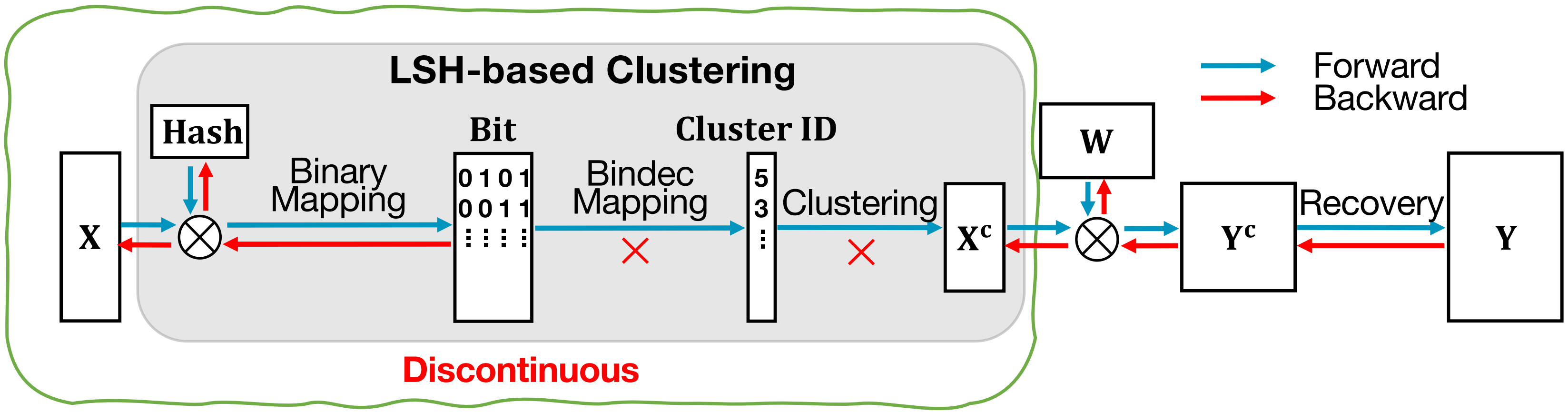
Bit **Transformation (H×2^H)**

$$\left(\begin{bmatrix}1 & 1\\0 & 0\\1 & 1\\1 & 1\end{bmatrix}+\begin{bmatrix}0 & 1\\0 & 1\\0 & 1\\0 & 1\end{bmatrix}\right)\times\begin{bmatrix}2^1/ & 2^1/ & 2^1/ & 2^1/\\2^0/ & 2^0/ & 2^0/ & 2^0/\end{bmatrix}=\begin{bmatrix}4 & 4 & 4 & 4\\1 & 1 & 1 & 1\\4 & 4 & 4 & 4\\4 & 4 & 4 & 4\end{bmatrix}$$

Quotient

$$\begin{bmatrix}4 & 4 & 4 & 4\\1 & 1 & 1 & 1\\4 & 4 & 4 & 4\\4 & 4 & 4 & 4\end{bmatrix}\times\begin{bmatrix}1/1 & 1/2 & 1/3 & 1/4\\1/1 & 1/2 & 1/3 & 1/4\end{bmatrix}=\begin{bmatrix}4 & 2 & 4/3 & 1\\1 & 1/2 & 1/3 & 1/4\\4 & 2 & 4/3 & 1\\4 & 2 & 4/3 & 1\end{bmatrix}$$

Backward Pass



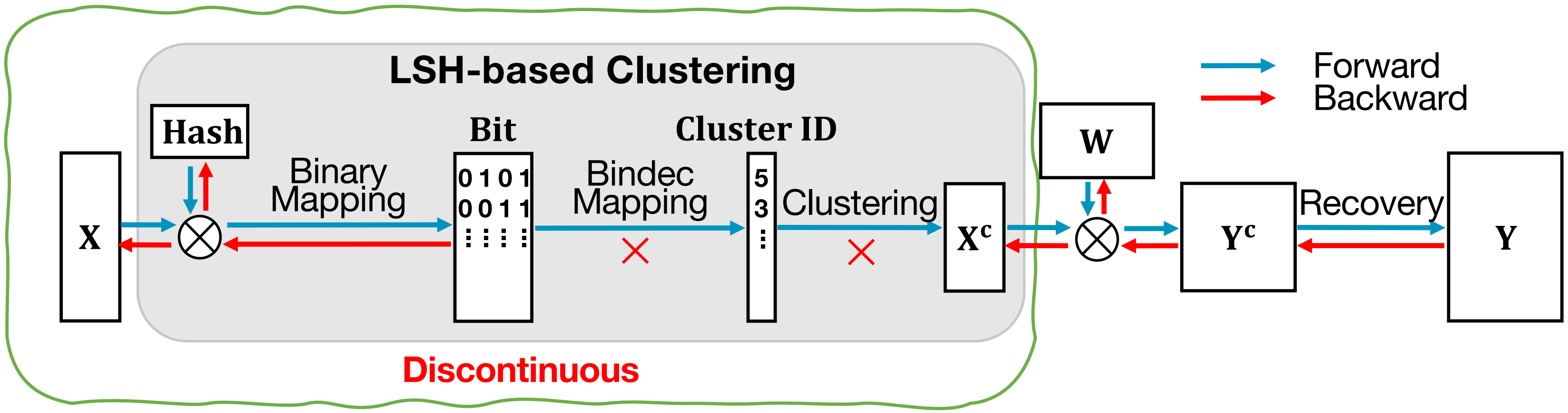
Bit, Transformation (H×2^H), Quotient

$$\left(\begin{bmatrix}1 & 1\\0 & 0\\1 & 1\\1 & 1\end{bmatrix}+\begin{bmatrix}0 & 1\\0 & 1\\0 & 1\\0 & 1\end{bmatrix}\right)\times\begin{bmatrix}2^1/1 & 2^1/2 & 2^1/3 & 2^1/4\\2^0/1 & 2^0/2 & 2^0/3 & 2^0/4\end{bmatrix}=\begin{bmatrix}4 & 2 & 4/3 & 1\\1 & 1/2 & 1/3 & 1/4\\4 & 2 & 4/3 & 1\\4 & 2 & 4/3 & 1\end{bmatrix}$$

Space waste

Renmin University of China
North Carolina State University

Backward Pass



Bit **Transformation ($H \times 2^H$)** **Quotient**

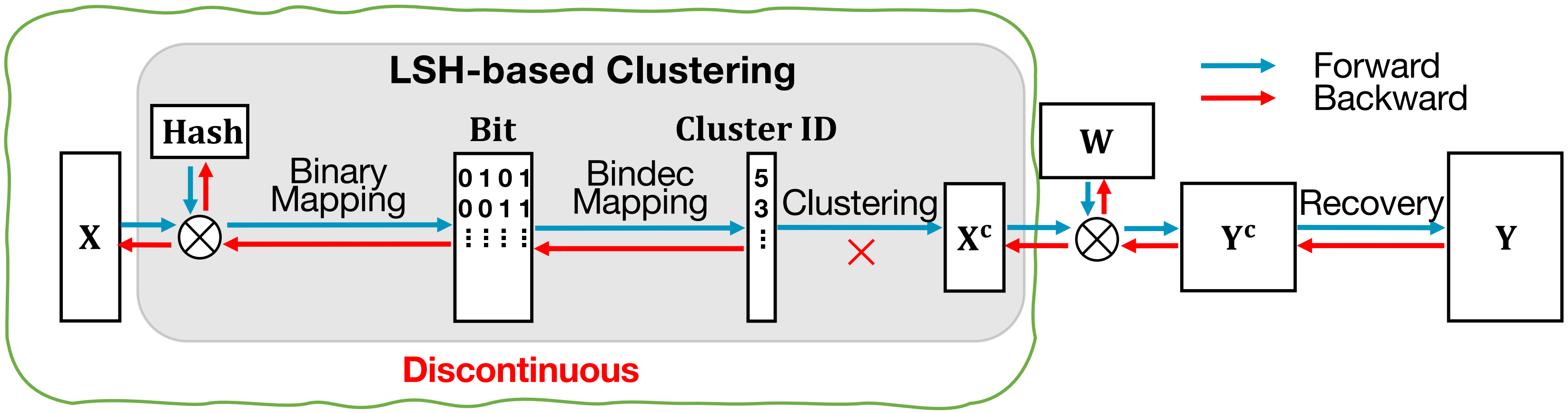
$$\left(\begin{bmatrix}1&1\\0&0\\1&1\\1&1\end{bmatrix}+\begin{bmatrix}0&1\\0&1\\0&1\\0&1\end{bmatrix}\right)\times\begin{bmatrix}2^1/1&2^1/2&2^1/3&2^1/4\\2^0/1&2^0/2&2^0/3&2^0/4\end{bmatrix}=\begin{bmatrix}4&2&4/3&1\\1&1/2&1/3&1/4\\4&2&4/3&1\\4&2&4/3&1\end{bmatrix}$$

Forward information

$$\left(\begin{bmatrix}1&1\\0&0\\1&1\\1&1\end{bmatrix}+\begin{bmatrix}0&1\\0&1\\0&1\\0&1\end{bmatrix}\right)\times\begin{bmatrix}2^1/1&2^1/4\\2^0/1&2^0/4\end{bmatrix}=\begin{bmatrix}4&1\\1&1/4\\4&1\\4&1\end{bmatrix}$$

Renmin University of China
North Carolina State University

Backward Pass



Bit · **Transformation (H×2^H)** · **Quotient**

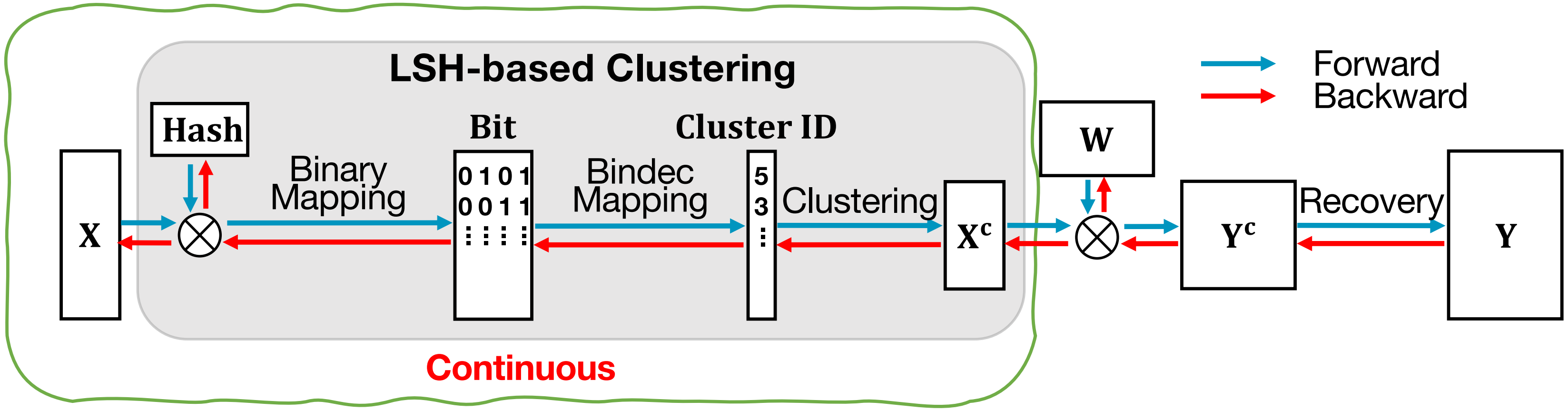
$$\left(\begin{bmatrix}1&1\\0&0\\1&1\\1&1\end{bmatrix}+\begin{bmatrix}0&1\\0&1\\0&1\\0&1\end{bmatrix}\right)\times\begin{bmatrix}2^1/1&2^1/2&2^1/3&2^1/4\\2^0/1&2^0/2&2^0/3&2^0/4\end{bmatrix}=\begin{bmatrix}4&2&4/3&1\\1&1/2&1/3&1/4\\4&2&4/3&1\\4&2&4/3&1\end{bmatrix}$$

Forward information · **Average (Avg)**

$$\left(\begin{bmatrix}1&1\\0&0\\1&1\\1&1\end{bmatrix}+\begin{bmatrix}0&1\\0&1\\0&1\\0&1\end{bmatrix}\right)\times\begin{bmatrix}2^1/1&2^1/4\\2^0/1&2^0/4\end{bmatrix}=\begin{bmatrix}4&1\\1&1/4\\4&1\\4&1\end{bmatrix}\xrightarrow{\text{Gaussian}}\begin{bmatrix}0&1\\1&0\\0&1\\0&1\end{bmatrix}\xrightarrow{\text{Divide } N_c}\begin{bmatrix}0&1/3\\1&0\\0&1/3\\0&1/3\end{bmatrix}$$

Renmin University of China
North Carolina State University

Backward Pass



- Clustering

$$\begin{array}{c}\mathbf{Avg}^T\\ \begin{bmatrix}1/3 & 0 & 1/3 & 1/3\\ 0 & 1 & 0 & 0\end{bmatrix}\end{array} \times \begin{array}{c}\mathbf{X}\\ \begin{bmatrix}1 & 2 & 3\\ 0 & 0 & 1\\ 0 & 2 & 1\\ 1 & 3 & 2\end{bmatrix}\end{array} = \begin{array}{c}\mathbf{X}^c\\ \begin{bmatrix}2/3 & 7/3 & 2\\ 2 & 1 & 1\end{bmatrix}\end{array}$$

Experiments

Platforms

- **Training:** A server machine with an Intel Core i7-12700K processor and an NVIDIA GeForce RTX A6000 GPU.
- **Inference:** An STM32F469NI MCU.

Datasets

- Cifar-10
- ImageNet-64x64

End-to-end Performance Comparison

Network	Conv Method	Average Time per Image (ms)	Top-1 Accuracy (%)
CifarNet	Conventional	217.32	78.2
	Deep Reuse	154.44	$73.2 \sim 76.1$
	TREC	153.92	76.5
ZfNet	Conventional	3557.32	80.1
	Deep Reuse	814.03	$72.5 \sim 76.6$
	TREC	814.01	78.9
Vanilla SqueezeNet	Conventional	1639.51	83.5
	Deep Reuse	328.97	$79.8 \sim 81.9$
	TREC	327.90	83.0
SqueezeNet + Complex Bypass	Conventional	1998.86	85.3
	Deep Reuse	543.71	$80.5 \sim 83.1$
	TREC	544.03	84.6
ResNet-34 (ImageNet-64×64)	Conventional	4242.77	52.6
	Deep Reuse	1379.26	$46.7 \sim 49.9$
	TREC	1378.75	52.2

Experiments

Benefits

- Speedup with minor accuracy loss.
- Stable performance and high accuracy.
- A plug-and-play replacement for convolutional layers in mainstream CNNs.
- Orthogonal to lasting redundancy elimination methods (e.g. pruning, quantization).

End-to-end Performance Comparison

Network	Conv Method	Average Time per Image (ms)	Top-1 Accuracy (%)
CifarNet	Conventional	217.32	78.2
	Deep Reuse	154.44	73.2 ~ 76.1
	TREC	153.92	76.5
ZfNet	Conventional	3557.32	80.1
	Deep Reuse	814.03	72.5 ~ 76.6
	TREC	814.01	78.9
Vanilla SqueezeNet	Conventional	1639.51	83.5
	Deep Reuse	328.97	79.8 ~ 81.9
	TREC	327.90	83.0
SqueezeNet + Complex Bypass	Conventional	1998.86	85.3
	Deep Reuse	543.71	80.5 ~ 83.1
	TREC	544.03	84.6
ResNet-34 (ImageNet-64×64)	Conventional	4242.77	52.6
	Deep Reuse	1379.26	46.7 ~ 49.9
	TREC	1378.75	52.2

Thanks for listening

guanjw@ruc.edu.cn