45th International Symposium on Computer Architecture, Los Angeles, US, June 2018

Computation Reuse in DNNs by Exploiting Input Similarity

Marc Riera, Jose Maria Arnau, Antonio González

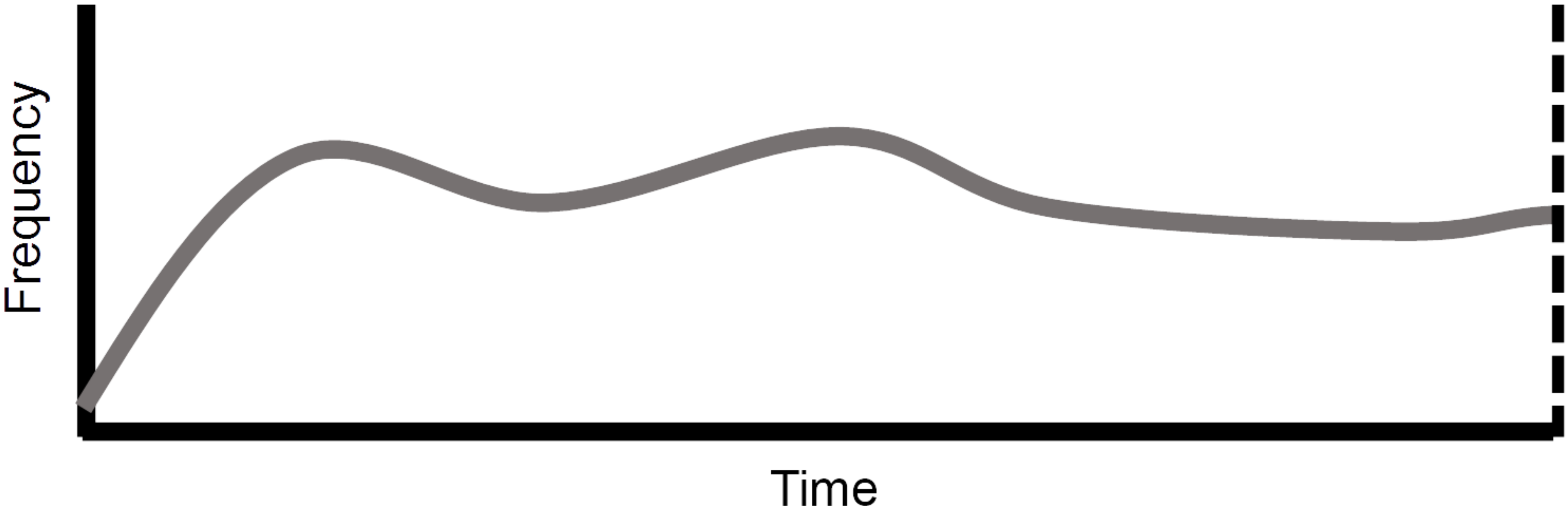
UPC

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

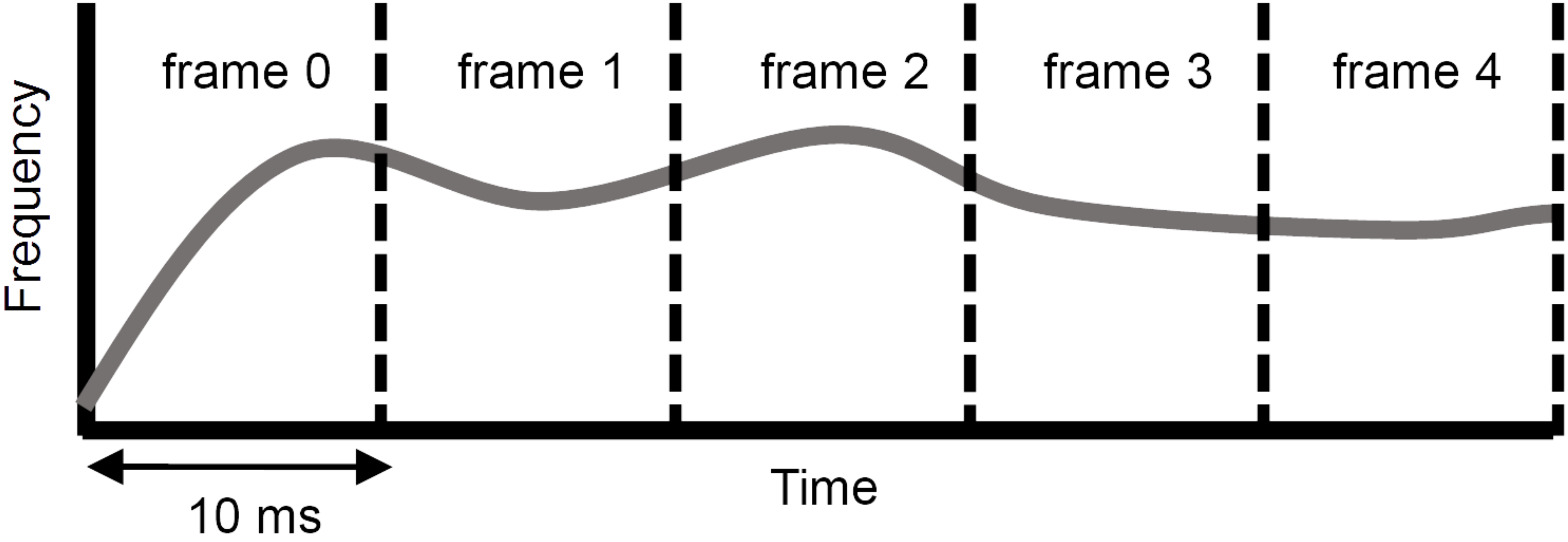
Departament d'Arquitectura de Computadors

Sequence Processing Applications

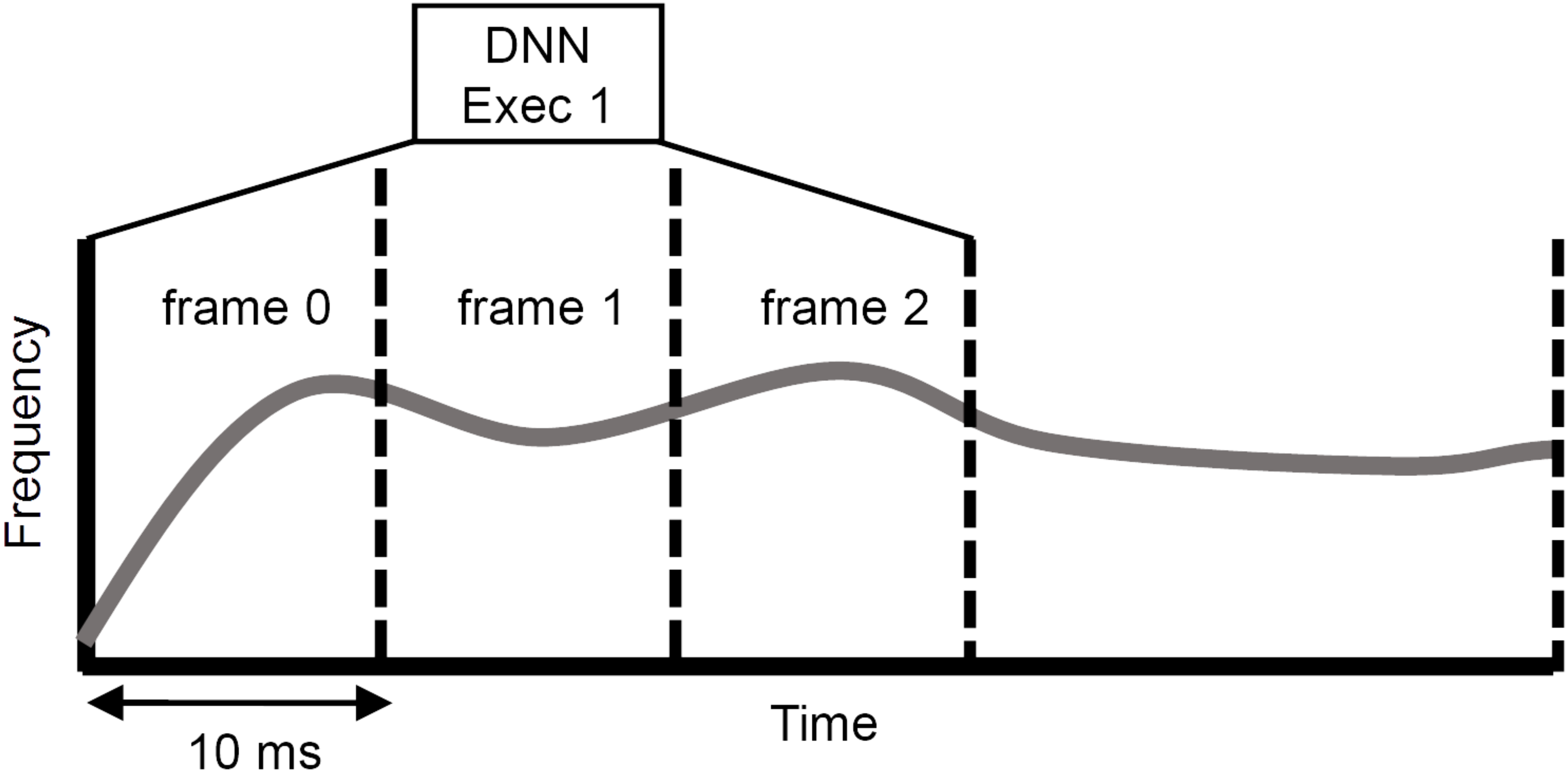
Speech Audio Signal



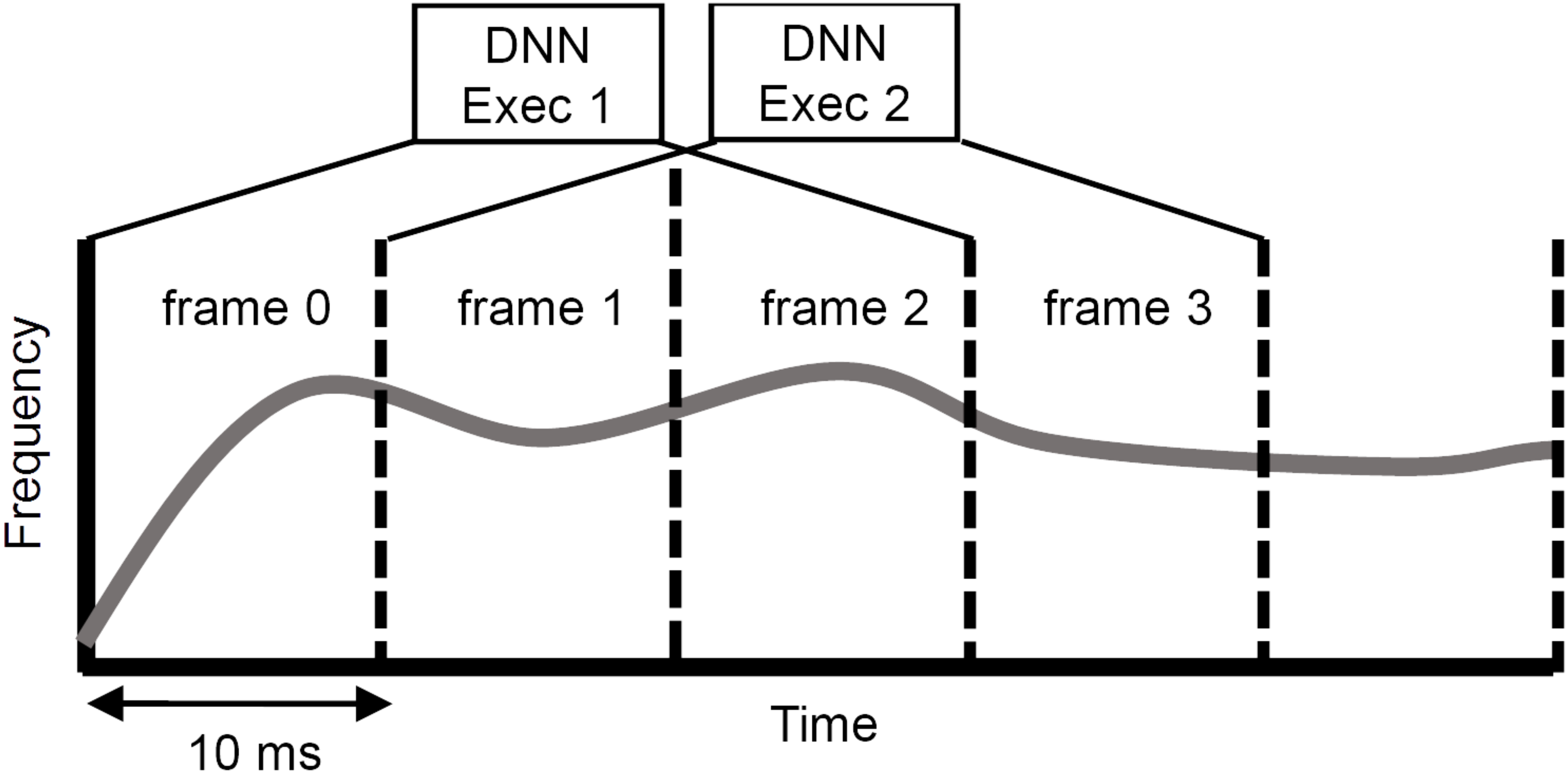
Sequence Processing Applications



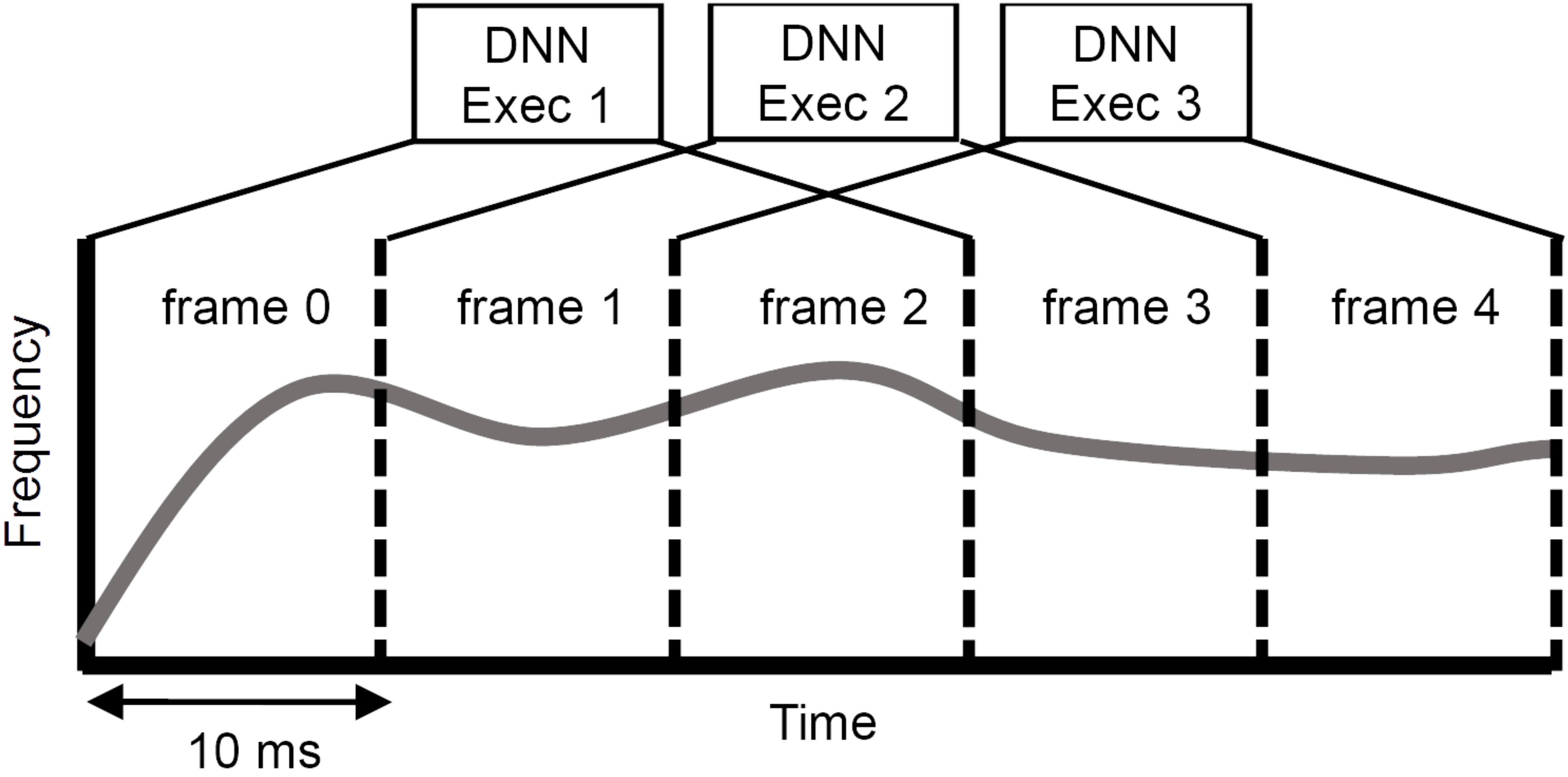
Sequence Processing Applications



Sequence Processing Applications



Sequence Processing Applications

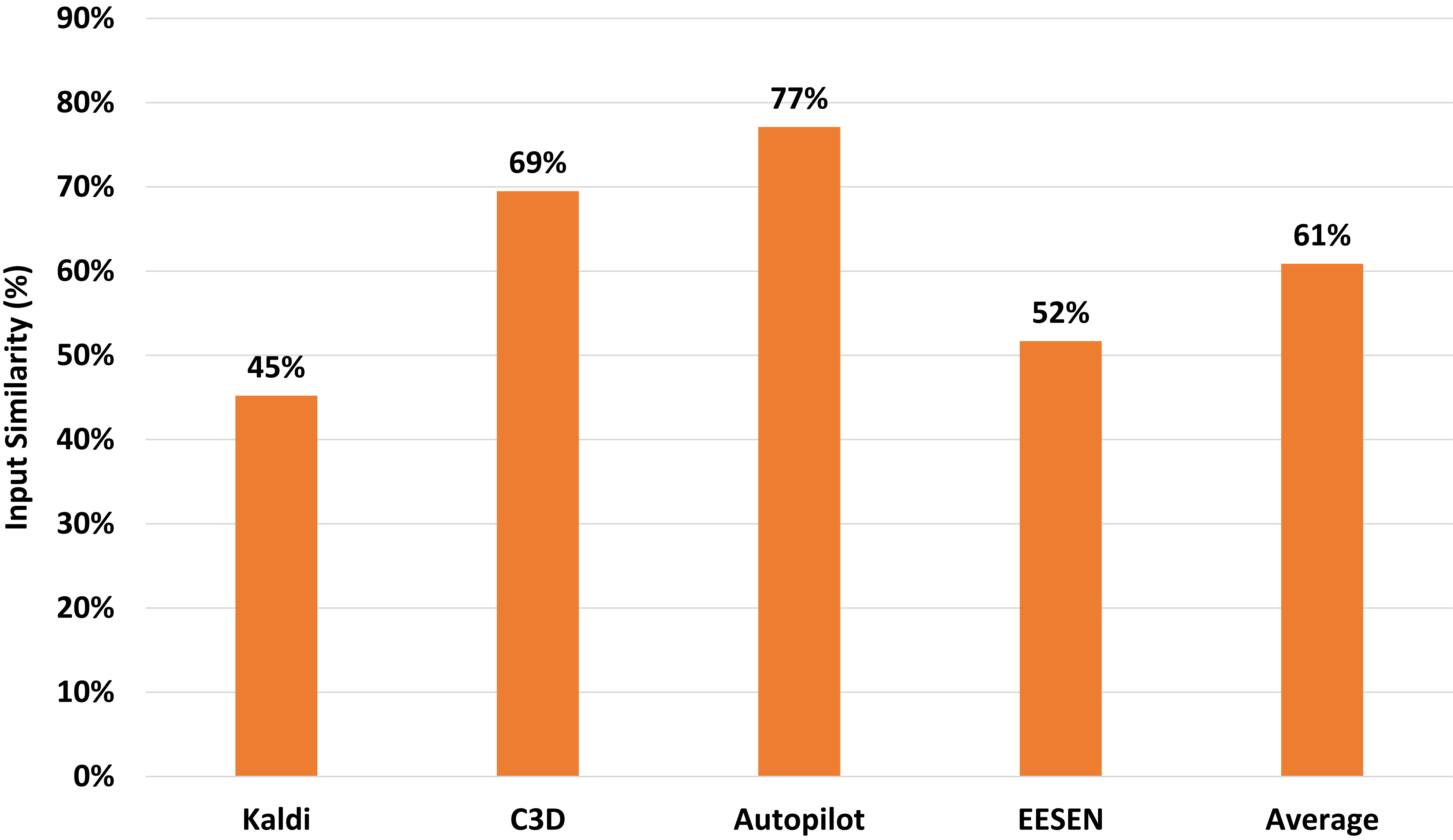


Speech Recognition DNN executions to classify a sequence of audio frames in phonemes

Benchmarks

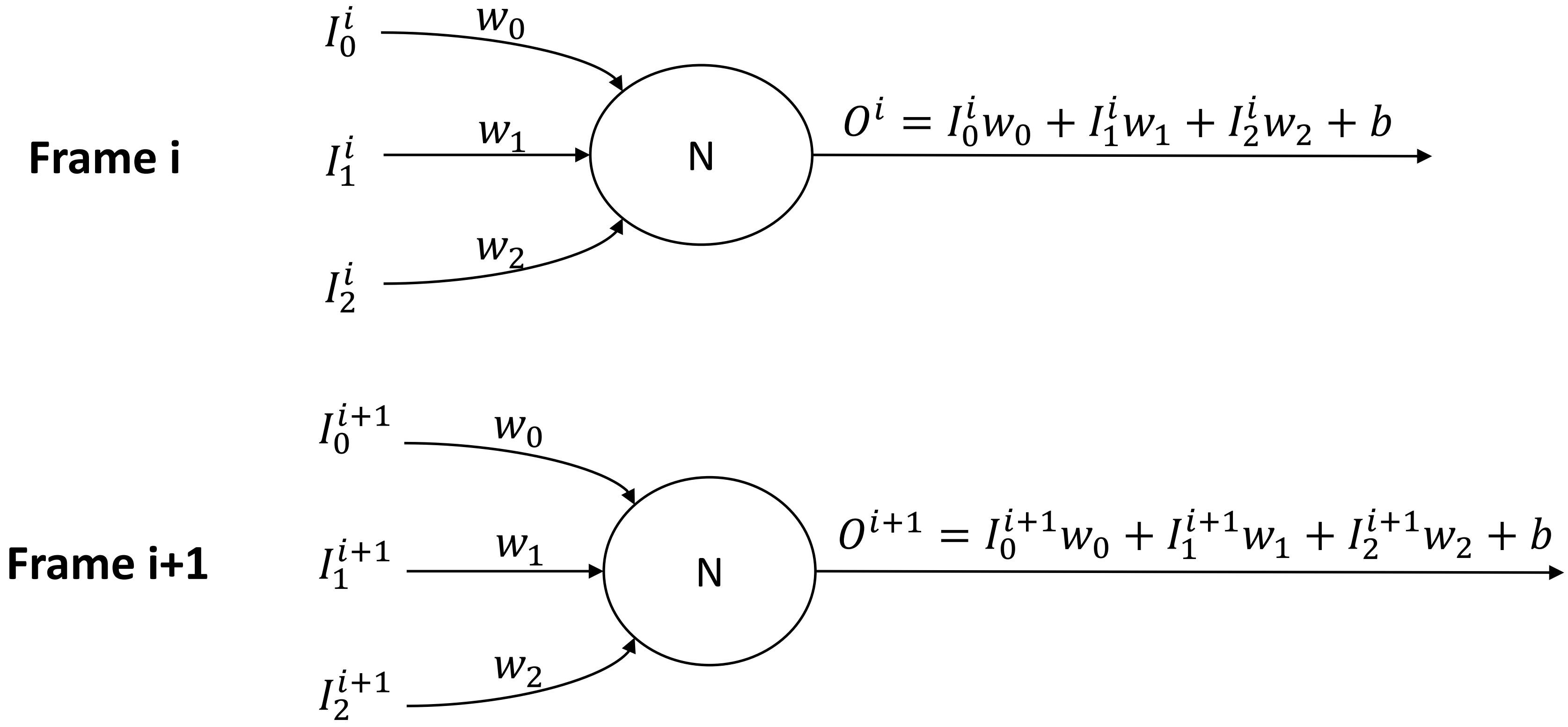
DNN Name	DNN Type	DNN Application	#Parameters	Accuracy
Kaldi	MLP	Acoustic Scoring	4,7M	89,04%
EESEN	RNN	Speech Recognition	11M	68,85%
C3D	CNN	Video Classification	78M	93,48%
AutoPilot	CNN	Self-Driving Cars	1,6M	99,63%

Input Similarity



Exploiting Temporal Similarity Example

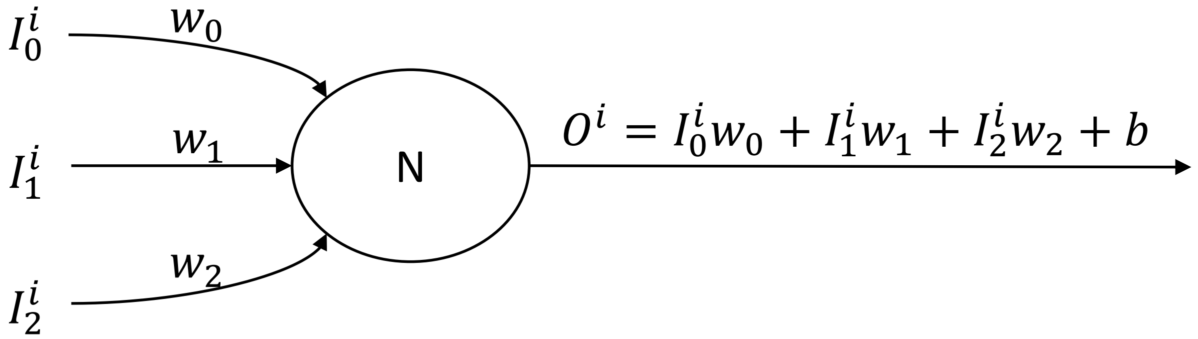
Baseline


Frame i
I_0^i
w_0
I_1^i
w_1
I_2^i
w_2
N
$O^i = I_0^i w_0 + I_1^i w_1 + I_2^i w_2 + b$
Frame i+1
I_0^{i+1}
w_0
I_1^{i+1}
w_1
I_2^{i+1}
w_2
N
$O^{i+1} = I_0^{i+1} w_0 + I_1^{i+1} w_1 + I_2^{i+1} w_2 + b$

Exploiting Temporal Similarity Example

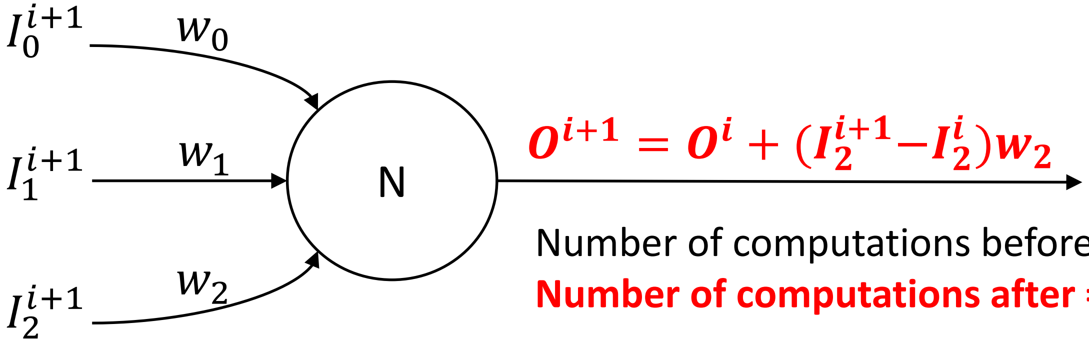
Proposal

Frame i



$$O^i = I_0^i w_0 + I_1^i w_1 + I_2^i w_2 + b$$

Frame i+1



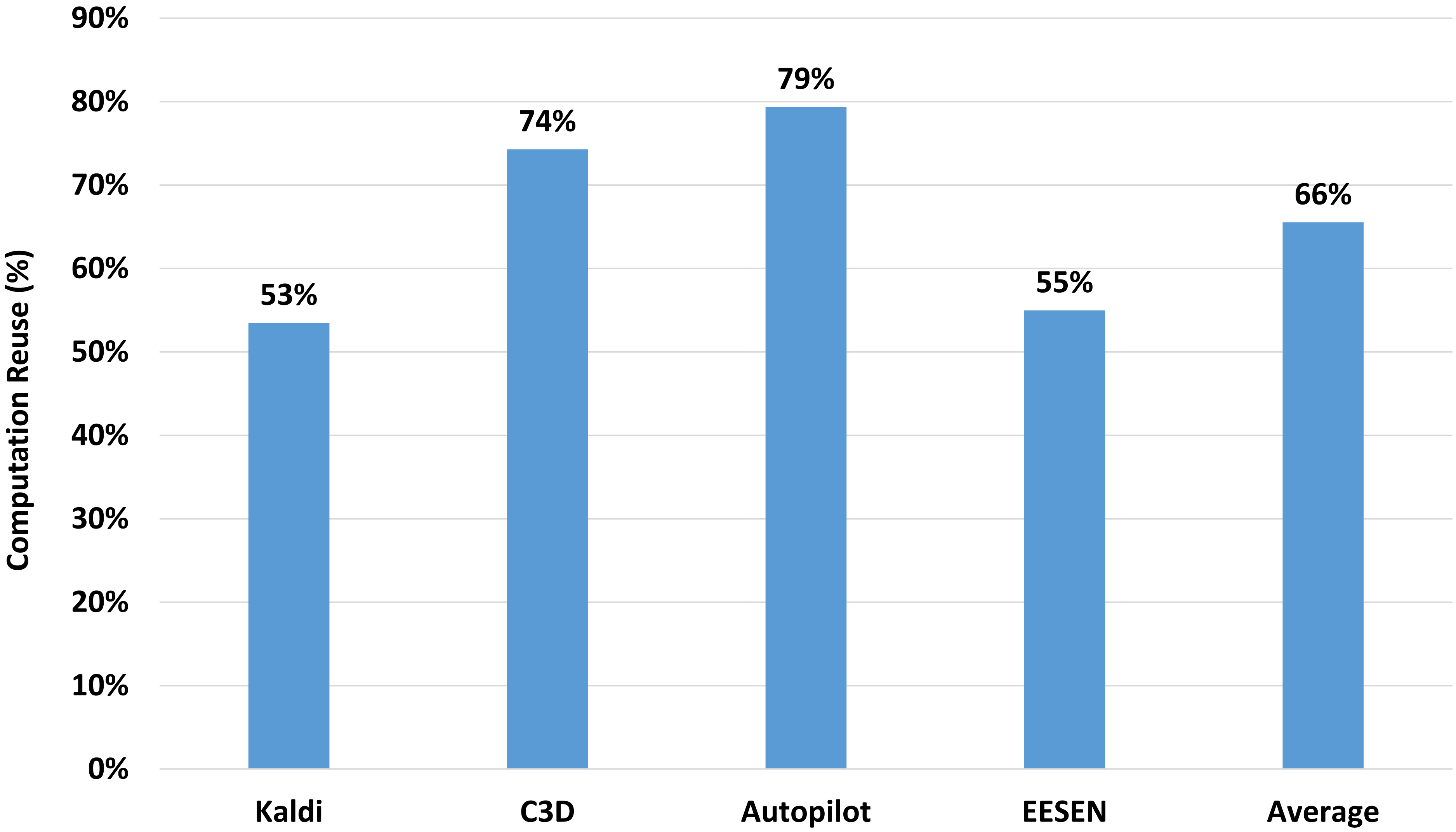
$$\boldsymbol{O^{i+1} = O^i + (I_2^{i+1} - I_2^i) w_2}$$

Number of computations before = 6

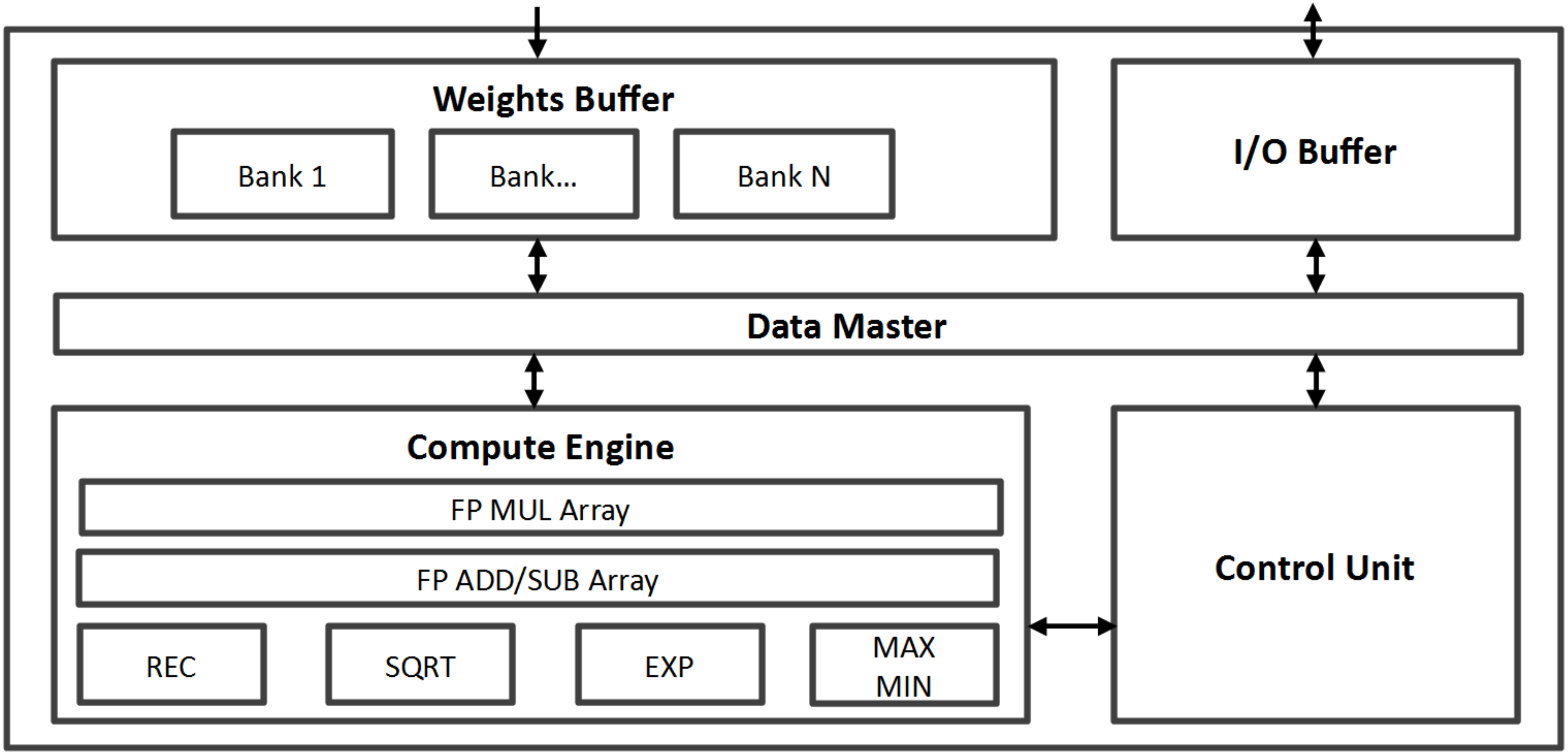
Number of computations after = 2

Note: Substraction of the inputs is almost negligible since its performed once per input

Computation Reuse

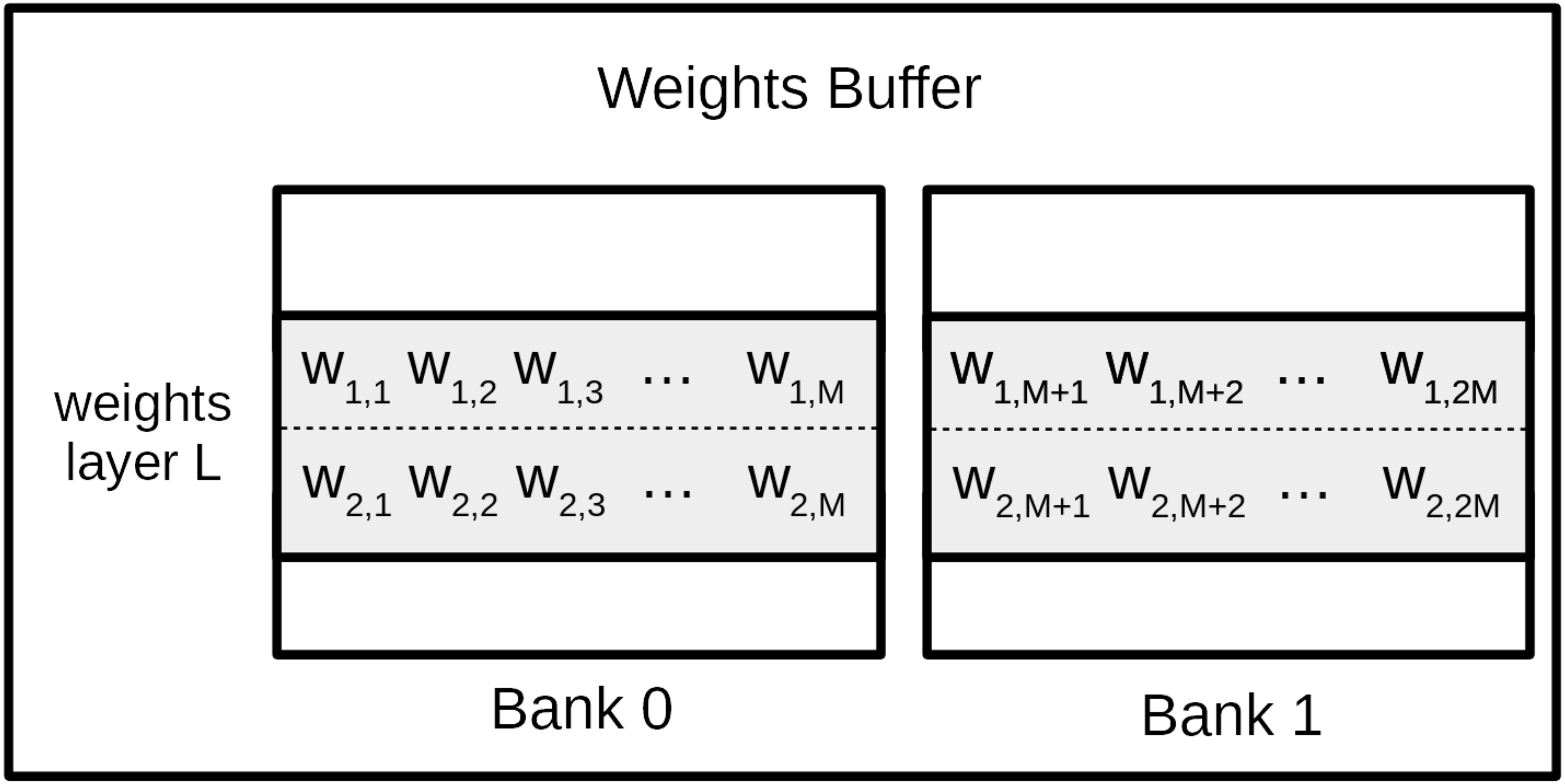


DNN Processing Unit



Tile

FC Execution in the Reuse Accelerator (1)



Weights Buffer

weights layer L

$W_{1,1}$ $W_{1,2}$ $W_{1,3}$ … $W_{1,M}$

$W_{2,1}$ $W_{2,2}$ $W_{2,3}$ … $W_{2,M}$

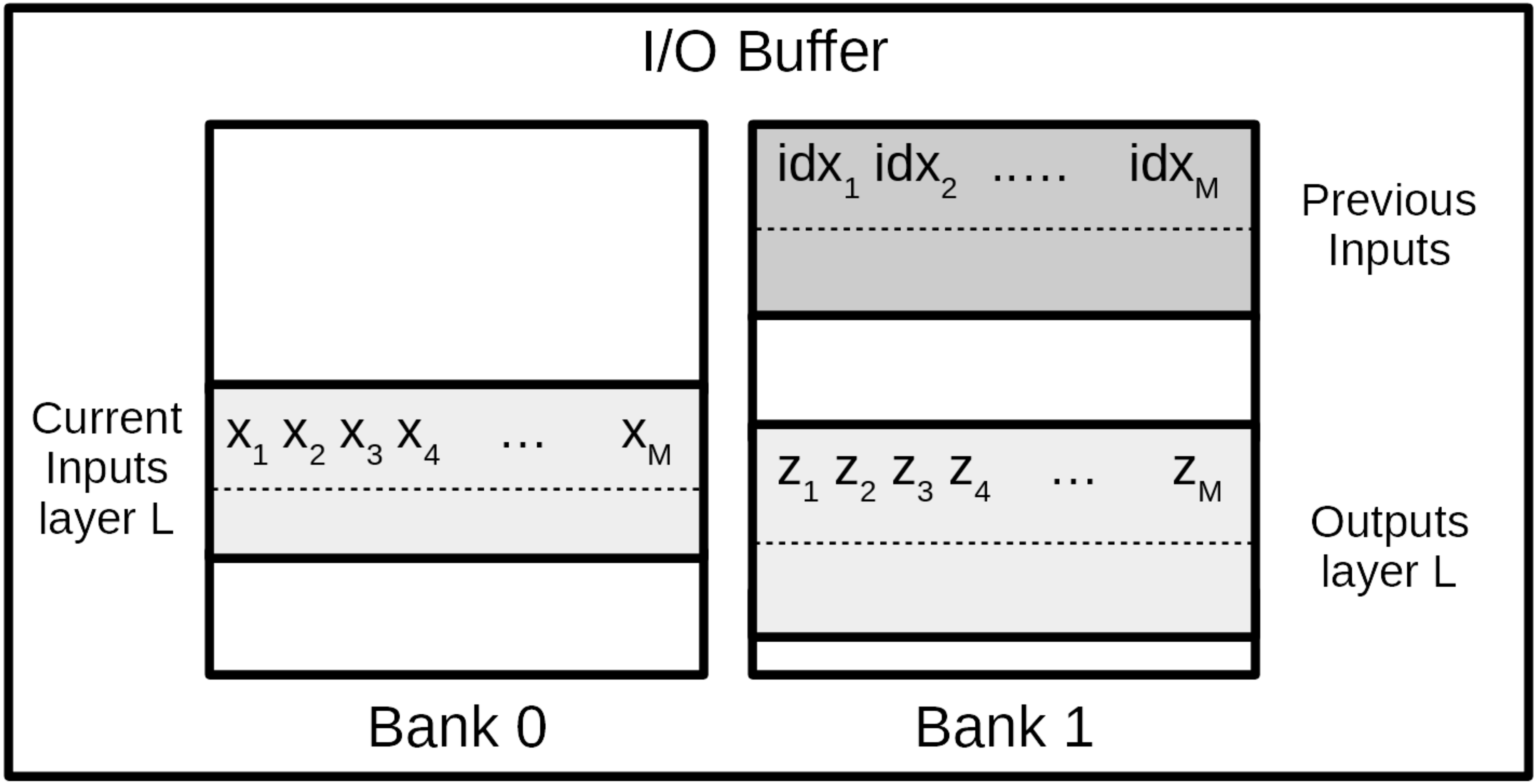
Bank 0

$W_{1,M+1}$ $W_{1,M+2}$ … $W_{1,2M}$

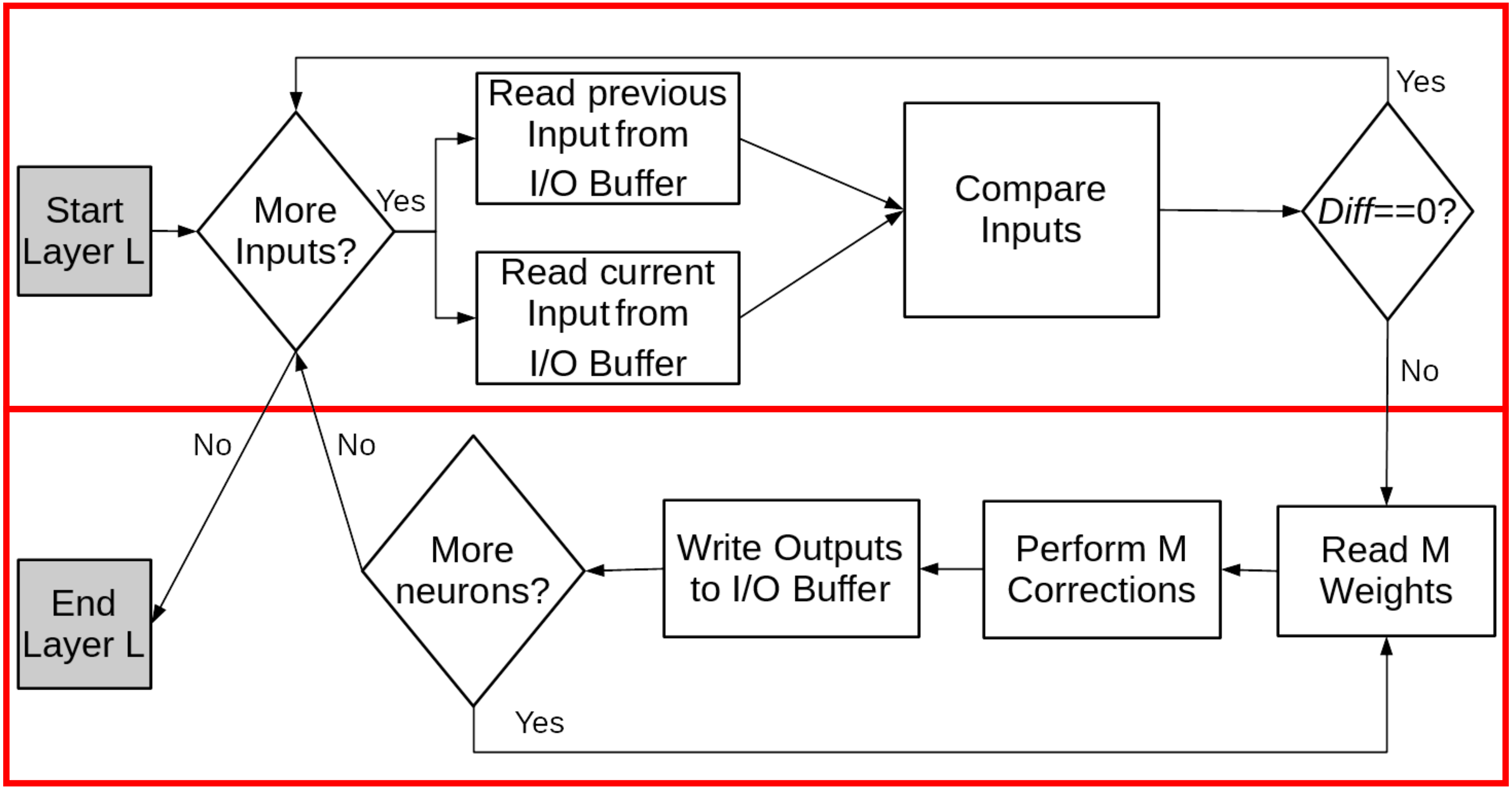
$W_{2,M+1}$ $W_{2,M+2}$ … $W_{2,2M}$

Bank 1

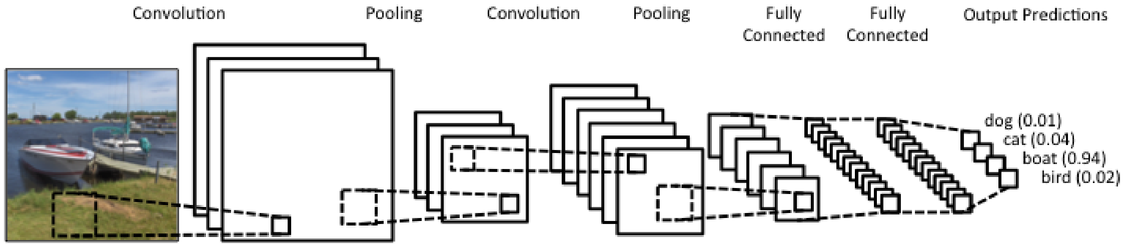
FC Execution in the Reuse Accelerator (2)


I/O Buffer
idx$_1$ idx$_2$ ..... idx$_M$
Previous Inputs
Current Inputs layer L
x$_1$ x$_2$ x$_3$ x$_4$ … x$_M$
z$_1$ z$_2$ z$_3$ z$_4$ … z$_M$
Outputs layer L
Bank 0
Bank 1

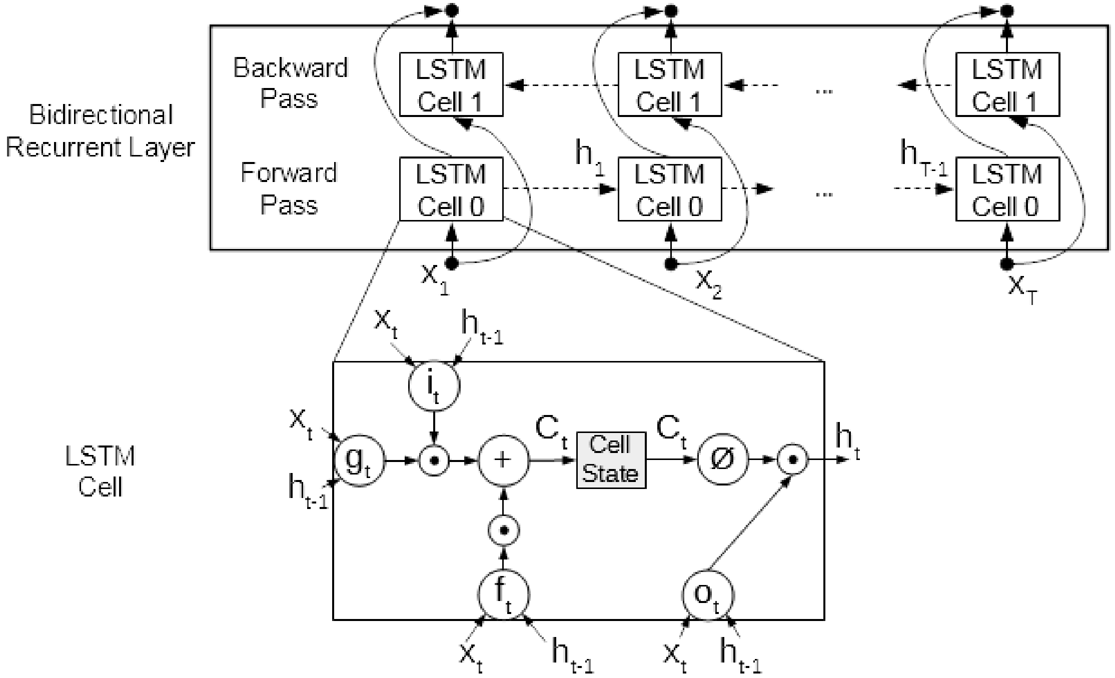
FC Execution in the Reuse Accelerator (3)



Other Supported Layers



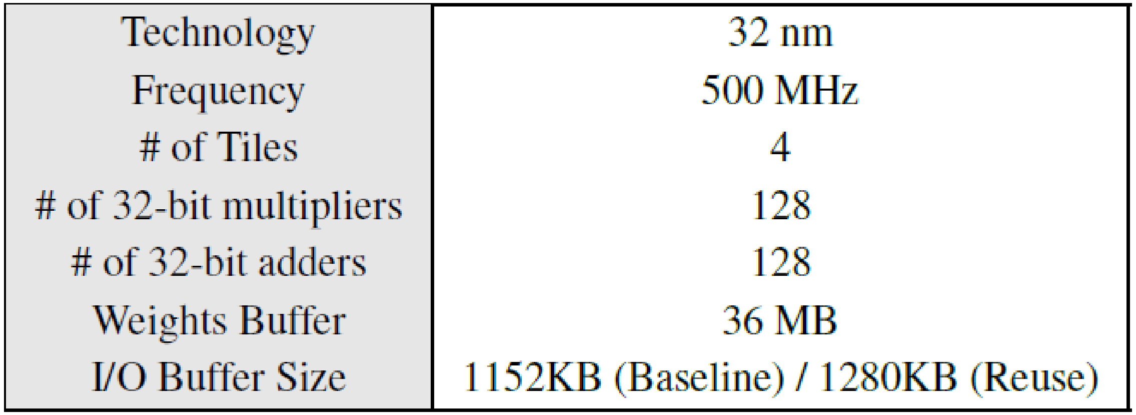
Convolutional Neural Network (CNN)



Recurrent Neural Network (RNN)

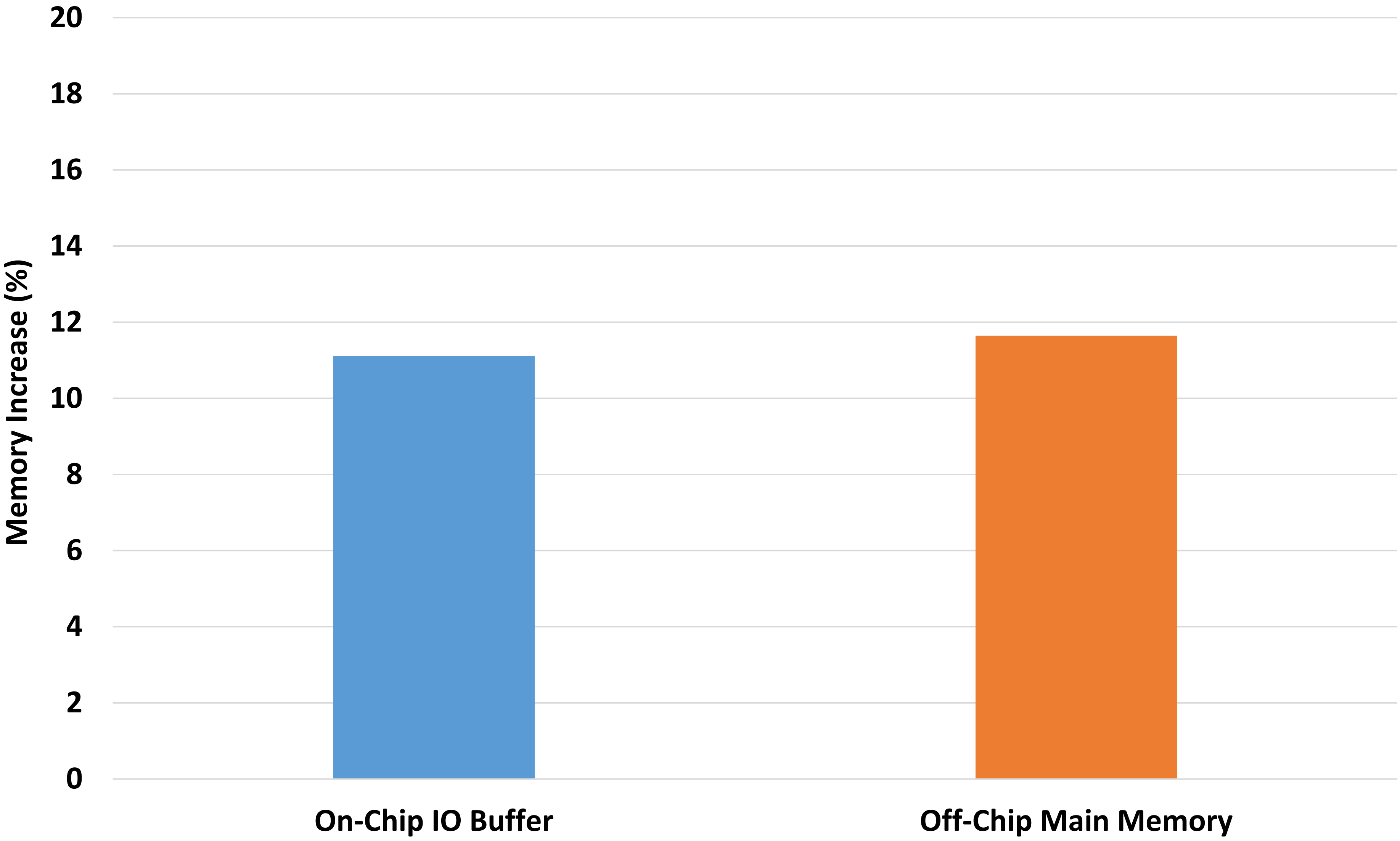
Evaluation Methodology

- Simulator to evaluate the performance and energy of the accelerator
- Design Compiler to obtain power and delay of logic modules
  - 28/32nm library from Synopsys and the DesignWare logic modules
- CACTI used for SRAM and eDRAM memories
- MICRON LPDDR4 for main Memory
- Accelerator Configuration:

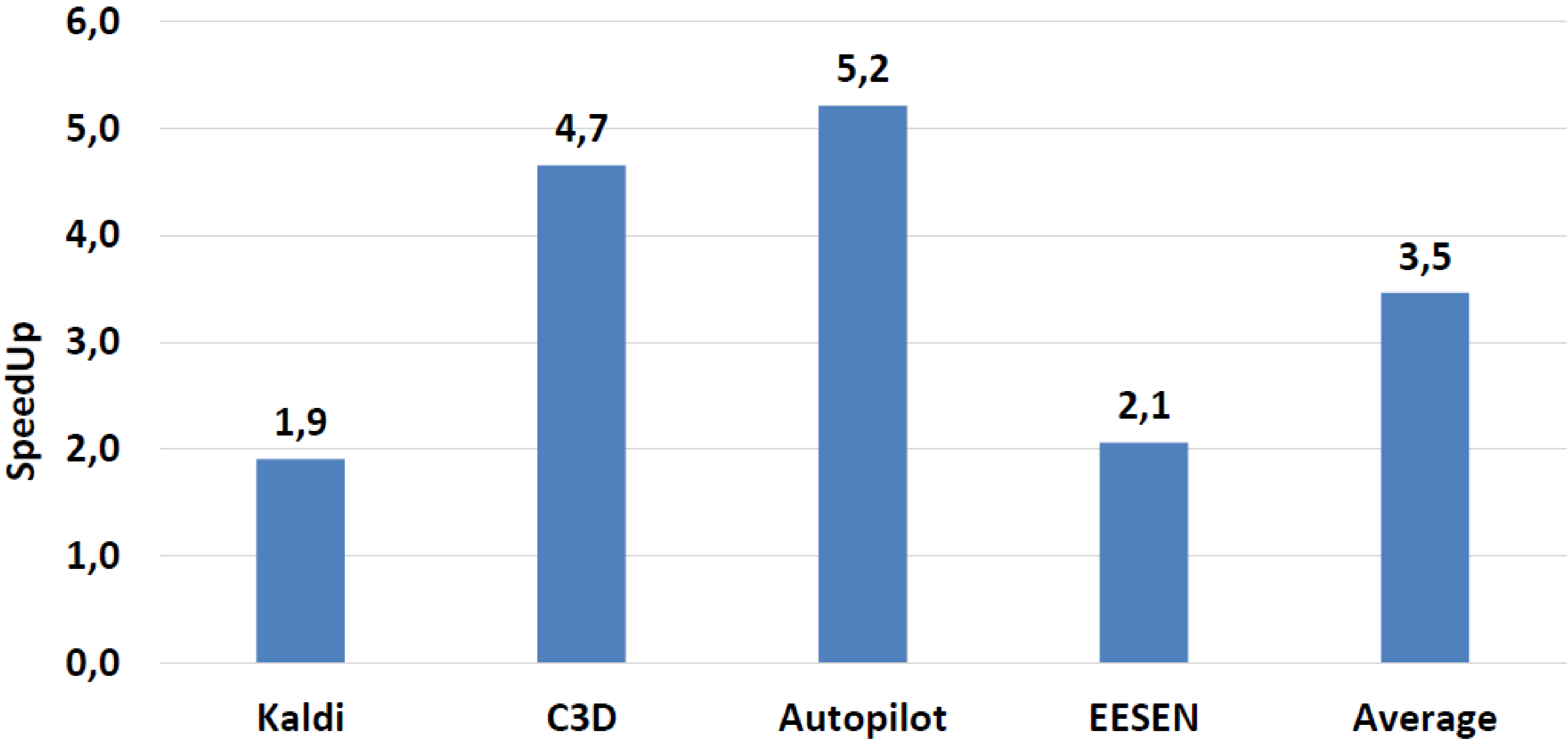


Technology	32 nm
Frequency	500 MHz
# of Tiles	4
# of 32-bit multipliers	128
# of 32-bit adders	128
Weights Buffer	36 MB
I/O Buffer Size	1152KB (Baseline) / 1280KB (Reuse)

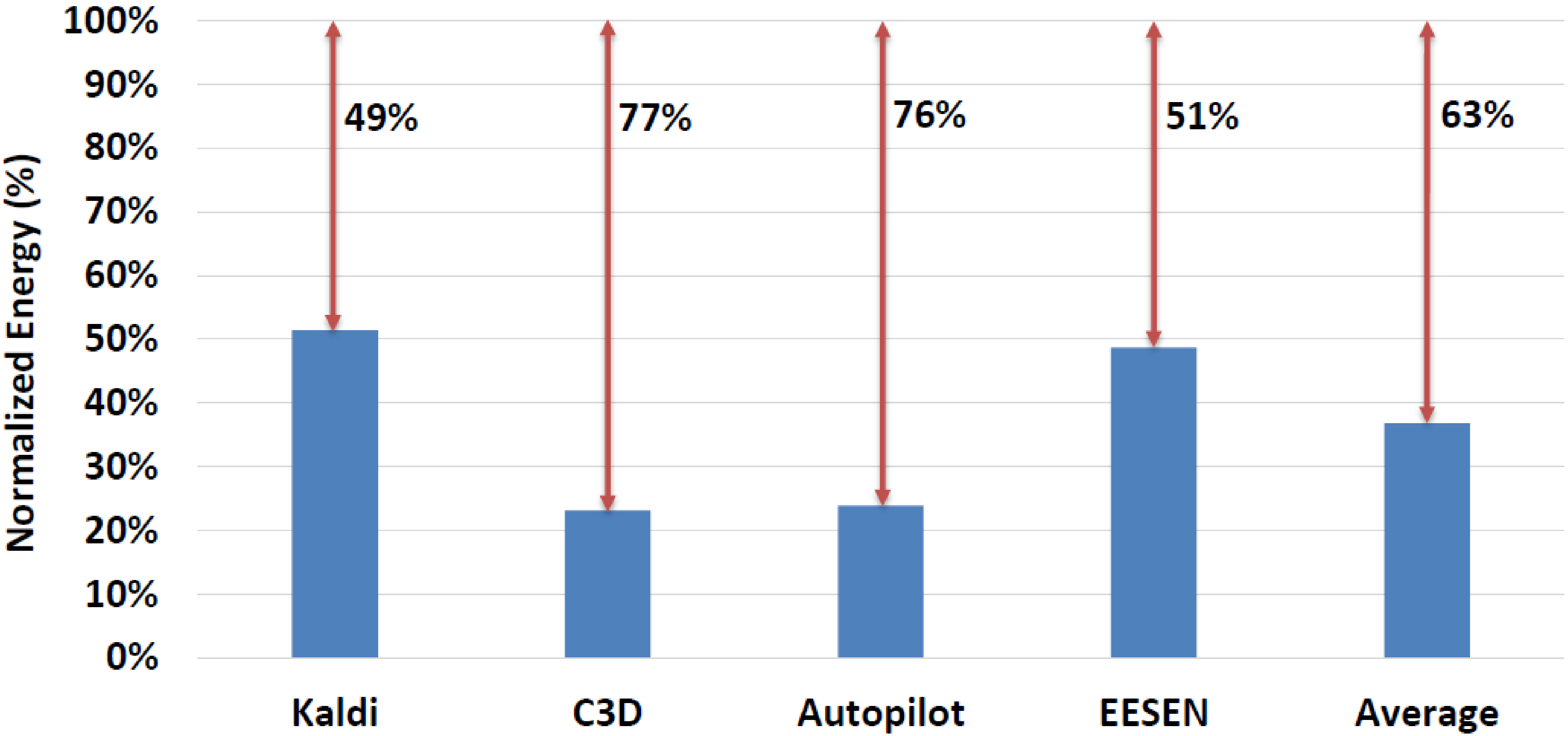
Memory Footprint Overheads



Results: SpeedUp



Results: Energy Savings



Conclusions

- More than 60% of the inputs remain unmodified respect the previous execution
- Our proposed scheme checks which inputs have changed:
  - Unmodified inputs are ignored, avoiding computations and memory accesses
  - Modified inputs are used to correct the previous output of each neuron
- On average, 63% energy savings and 3.5x speedup
- Small area overhead of less than 1% mainly for additional storage

45th International Symposium on Computer Architecture, Los Angeles, US, June 2018

Computation Reuse in DNNs by Exploiting Input Similarity

Marc Riera, Jose Maria Arnau, Antonio González

UPC

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Departament d'Arquitectura de Computadors