TENET: A Framework for Modeling Tensor Dataflow Based on Relation-centric Notation

Proceedings of ISCA'21

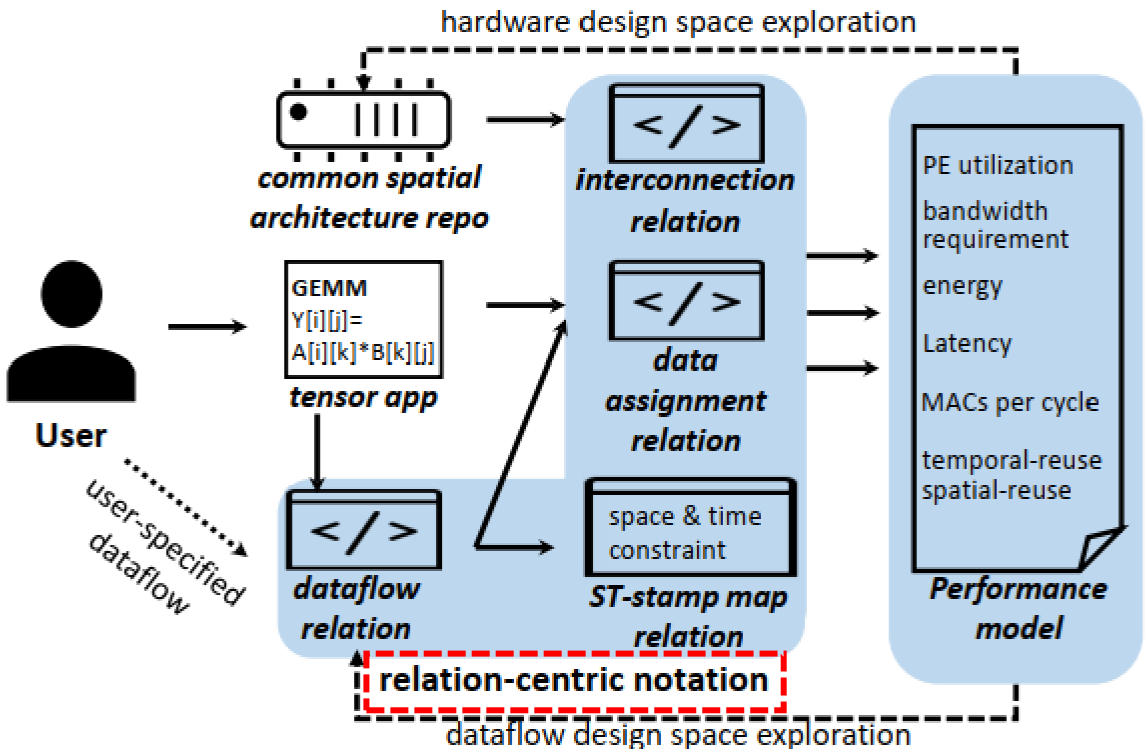
Lu, Liqiang, et al.
Peking University

Deep Learning Compiler Study
Dec 09, 2021

Presenter : Hyunwoo Cho
(hyunwoocho@sogang.ac.kr)

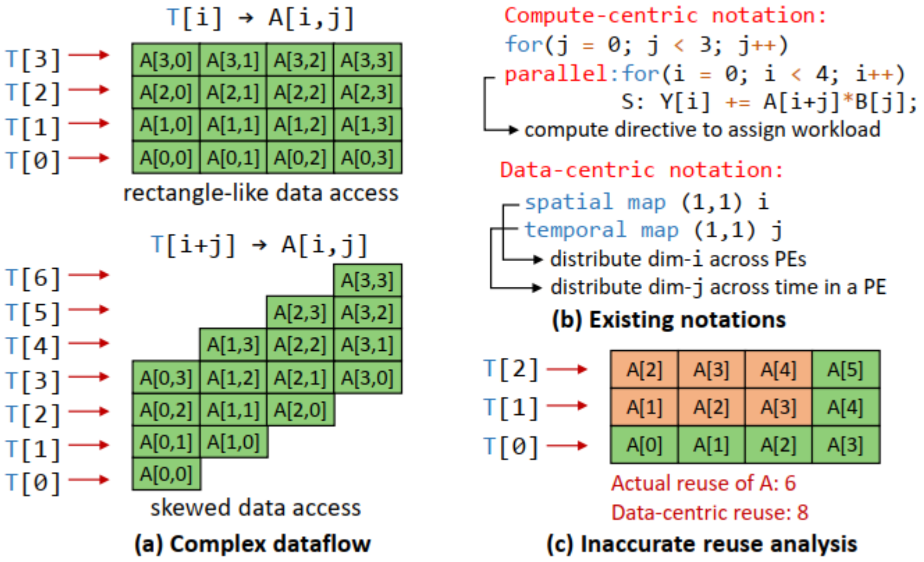
Introduction

- Accelerating tensor applications on spatial architectures provides high performance and energy-efficiency, but requires accurate performance models for evaluating various dataflow alternatives.
- This paper proposed a framework TENET that models hardware dataflow of tensor applications.



Problem

Features		Computation-centric Timeloop [39]	Computation-centric Interstellar [56]	Data-centric MAESTRO [24, 25]	STT [4, 9, 28, 54]	Relation-Centric TENET
Express dataflow	Instance execution sequence	loop order	loop order	temporal maps sequence of maps	time-stamp vector	multi-dim time-stamp
	PE workload assignment	`parallel` directive	`unroll` primitives	spatial maps	space-stamp matrix	multi-dim space-stamp
	Affine loop transformation	×	×	×	✓	✓
	Spatial architectures	✓	✓	✓	×	✓
Performance modeling	PE interconnection	×	×	×	×	✓
	Precise reuse analysis	×	×	×	×	✓
	Data assignment analysis	×	✓	✓	×	✓
	Bandwidth analysis	×	✓	✓	×	✓
	Latency / energy modeling	✓	×	✓	×	✓
	General tensor apps	×	×	×	✓	✓

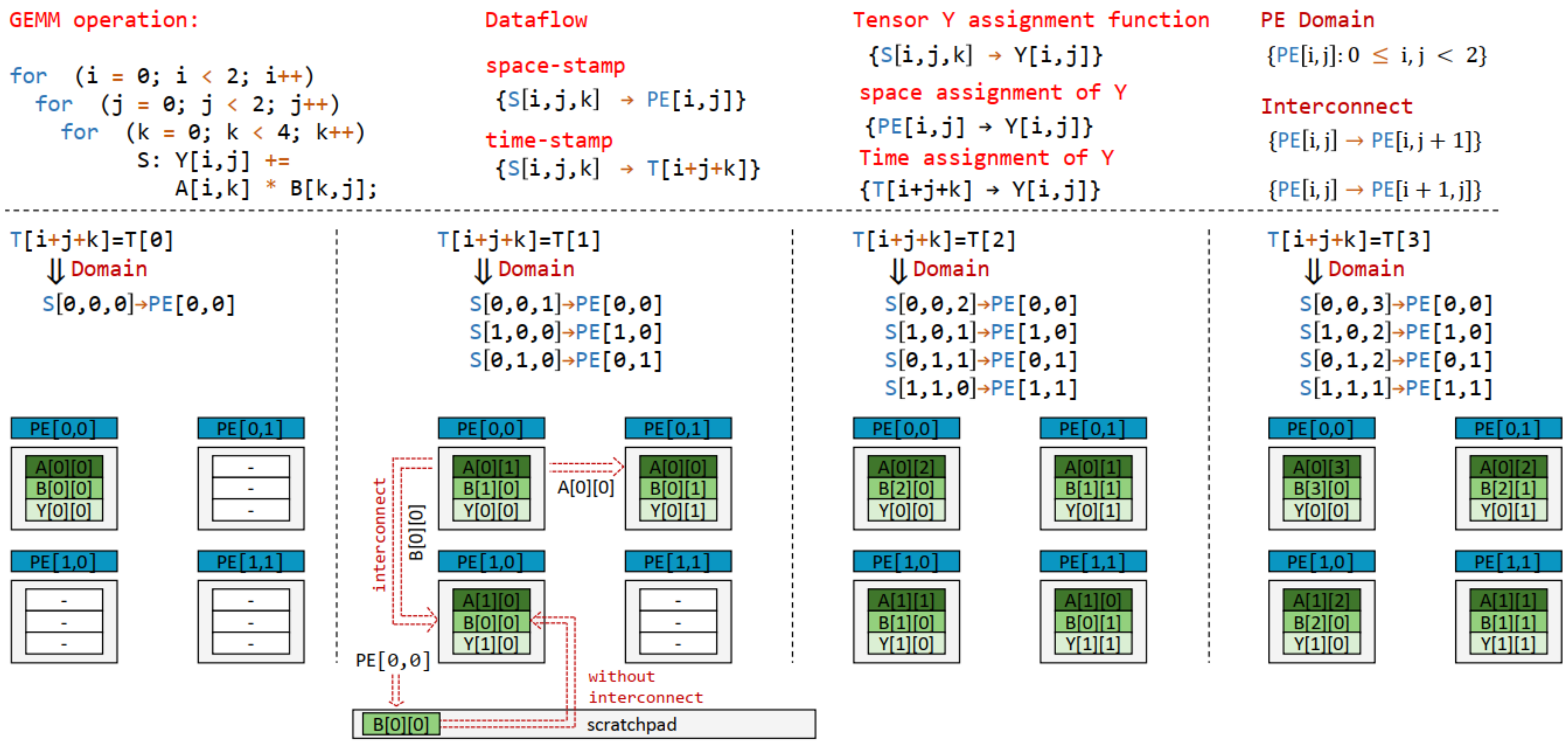


Requires additional affine transformation

- Limitations
  - Both computation-centric, data-centric notations are less expressive and they can only represent a subset space of hardware dataflows. Using these notations, architects are provided with an incomplete space and limited optimization opportunities.
  - Both notations fail to cover a complete design space of dataflow. For example, in Figure 1(a), we use T[t] to denote the tensor elements that are processed in cycle t. These two notations can only describe dataflows using rectangle-like data access, lacking the support for complex dataflows with skewed data access.
  - From performance modeling perspective, previous compute-centric notation-based models only analyze data reuse opportunities in a coarse-grained manner

TENET : Relation-centric notations

- Dataflow Relation : mapping loop instances onto PE array
- Data Assignment Relation : data assignment
- Interconnection Relation : interconnection between PE arrays
- Spacetime-stamp Map Relation : mapping between different spacetime-stamps



TENET : Relation-centric notations

- Dataflow Relation : mapping loop instances onto PE array

Loop instance Time-stamp

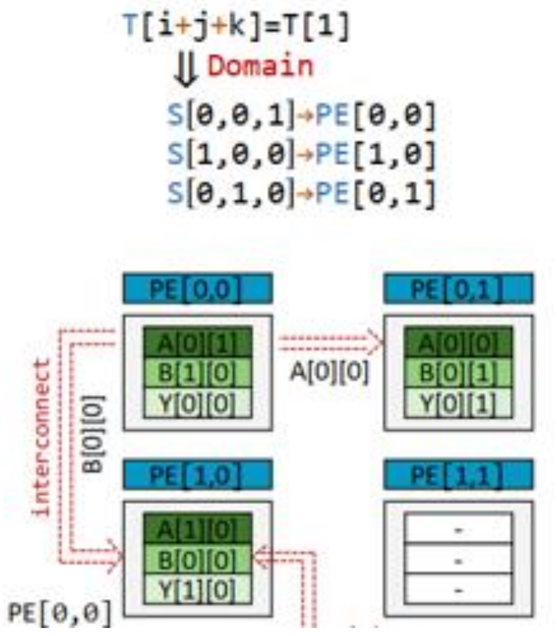
$$\Theta_{S,D} = \{S[\vec{n}] \rightarrow (PE[\vec{p}] \mid T[\vec{t}])\}, \quad S[\vec{n}] \in Ds$$

Space-stamp

MAESTRO TENET

$$O\left(n!\binom{n}{2}\right) \Rightarrow O\left(2^{n^2}\right)$$

Enlarged design space



$$\Theta_{S,D} = \{S[i,j,k] \rightarrow (PE[i,j] \mid T[i+j+k])\}$$

$$i+j+k=1$$ Time-stamp T[1]

$$constraint: 0 \le i,j < 2, \quad 0 \le k < 4$$

$$loop\ instances: [i,j,k] = [0,0,1],[1,0,0],[0,1,0]$$

TENET : Relation-centric notations

➢ Data Assignment Relation : data assignment

Given dataflow

$$A_{D,F} = \Theta_{S,D}^{-1} . A_{S,F} = \{(PE[\vec{p}] \mid T[\vec{t}]) \to F[\vec{f}]\}$$

Assign function

GEMM operation:

```
for  (i = 0; i < 2; i++)
  for  (j = 0; j < 2; j++)
    for  (k = 0; k < 4; k++)
       S: Y[i,j] +=
          A[i,k] * B[k,j];
```

Dataflow

space-stamp

{S[i,j,k] → PE[i,j]}

time-stamp

{S[i,j,k] → T[i+j+k]}

Tensor Y assignment function

{S[i,j,k] → Y[i,j]}

space assignment of Y

{PE[i,j] → Y[i,j]}

Time assignment of Y

{T[i+j+k] → Y[i,j]}

➢ Interconnection Relation : interconnection between PE arrays

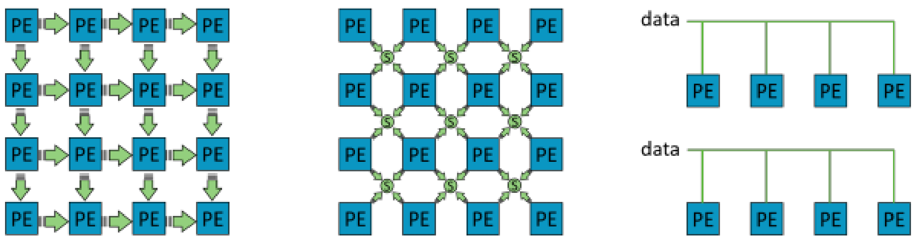
$$I_{PE_1,PE_2} = \{PE[\vec{p_1}] \to PE[\vec{p_2}]\} \ \ : c_1, \cdots, c_k$$

Interconnection : $\{\mathrm{PE[i,j]} \to \mathrm{PE[i',j']}\}$

2D-systolic : $(i' = i, j' = j+1) \ or \ (i' = i+1, j' = j)$

Mesh : $abs(i' - i) \leq 1 \ and \ abs(j' - j) \leq 1$

1D-Multicast : $abs(i' - i) \leq 3 \ \ (4 \ PEs)$



(a) 2D systolic (b) Mesh NoC (c) Multicast

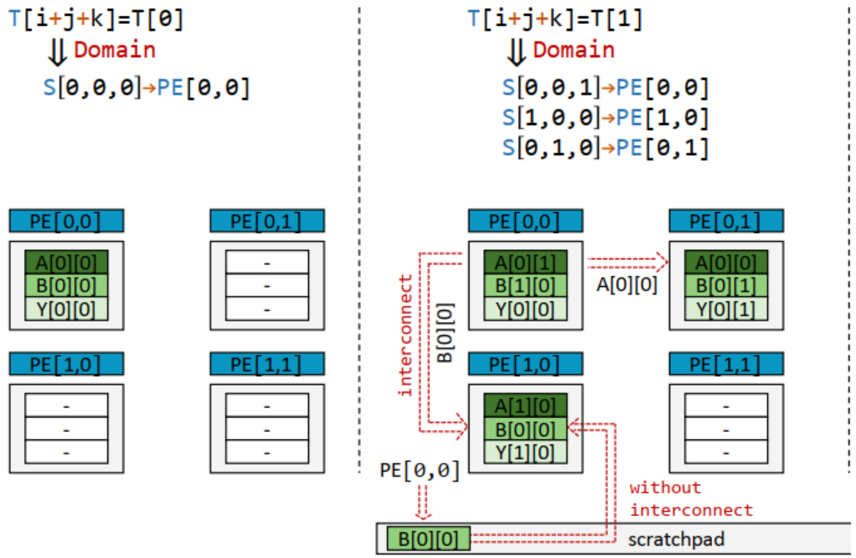
TENET : Relation-centric notations

- Spacetime-stamp Map Relation : mapping between different spacetime-stamps
  - By using data assignment and interconnections relations, TENET can correlate different spacetime-stamps based on the accessed data elements and their movement.

$$M_{D,D'} = \{([PE[\vec{p}_1] \mid T[\vec{t}_1]]) \rightarrow ([PE[\vec{p}_2] \mid T[\vec{t}_2]])\}$$

Spacetime set
D to D'

  - Spacetime-stamp relation can model the hardware behavior in continuous space-stamps and time-stamps.
  - Spacetime-stamp relation can detect data reuse both spatially and temporally.



map 1. $([PE[0,0] \mid T[0]) \rightarrow ([PE[0,0] \mid T[1]])$
map 2. $([PE[0,0] \mid T[0]) \rightarrow ([PE[0,1] \mid T[1]])$
map 3. $([PE[0,0] \mid T[0]) \rightarrow ([PE[1,0] \mid T[1]])$

A[0,0] traverses horizontally, B[0,0] traverses vertically

map 1.
$([PE[0,0] \mid \boxed{T[0]) \rightarrow Y[0,0]}$
$([PE[0,0] \mid \boxed{T[1]) \rightarrow Y[0,0]}$

Tensor element Y[0,0] is reused in the same PE.

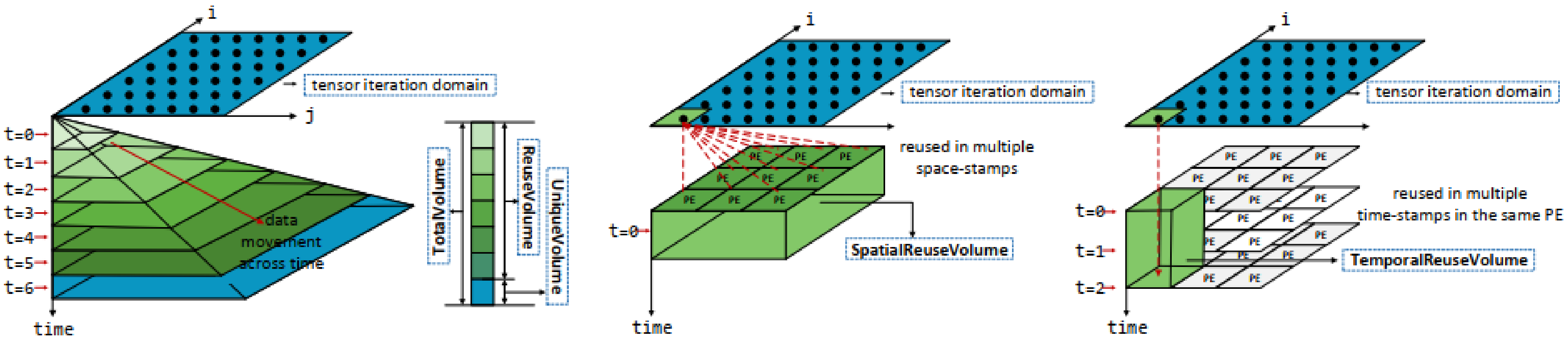
map 2.
$([PE[0,0] \mid \boxed{T[0]) \rightarrow A[0,0]}$
$([PE[0,1] \mid \boxed{T[1]) \rightarrow A[0,0]}$

map 3.
$([PE[0,0] \mid \boxed{T[0]) \rightarrow B[0,0]}$
$([PE[1,0] \mid \boxed{T[1]) \rightarrow B[0,0]}$

TENET : Performance model

➢ Data Reuse and Volume Model

- ➢ Total Volume : the total number of the tensor data accesses through the entire spacetime-stamp.
- ➢ Reuse Volume : the number of reused data across multiple spacetime-stamps.
- ➢ Unique Volume : the number of unique tensor data that are accessed.
- ➢ Spatial Reuse Volume : the amount of data reuse across multiple space-stamps.
- ➢ Temporal Reuse Volume : the amount of data reuse across multiple time-stamps within the same PE.

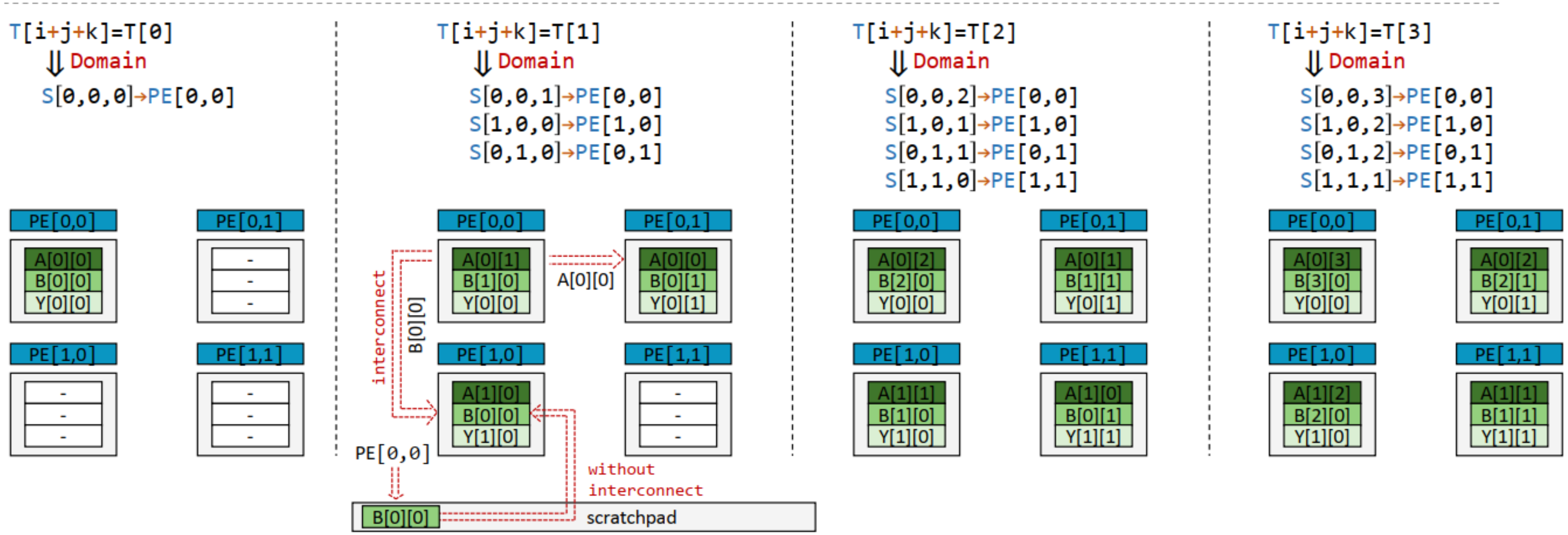


(a) TotalVolume, ReuseVolume, UniqueVolume (b) SpatialReuseVolume (c) TemporalReuseVolume

TENET : Performance model

➢ Data Reuse and Volume Model

- Total Volume : the total number of the tensor data accesses through the entire spacetime-stamp.
- Reuse Volume : the number of reused data across multiple spacetime-stamps.
- Unique Volume : the number of unique tensor data that are accessed.



time-stamp 0. used data $A[0][0]$
time-stamp 1. used data $A[0][1],\ A[0][0]\ A[1][0]$
time-stamp 2. used data $A[0][2],\ A[0][1],\ A[1][1],\ A[1][0]$
time-stamp 3. used data $A[0][3],\ A[0][2],\ A[1][2],\ A[1][1]$

$\boxed{\textbf{TotalVolume} = 1+3+4+4 = 12}$

time-stamp 1. reused data from time-stamp 0 $A[0][0]$
time-stamp 2. reused data from time-stamp 1 $A[0][1],\ A[1][0]$
time-stamp 3. reused data from time-stamp 2 $A[0][2],\ A[1][1]$

$\boxed{\textbf{ReuseVolume} = 1+2+2 = 5}$

time-stamp 0. new data $A[0][0]$
time-stamp 1. new data $A[0][1],\ A[1][0]$
time-stamp 2. new data $A[0][2],\ A[1][1]$
time-stamp 3. new data $A[0][3],\ A[1][2]$

$\boxed{\textbf{UniqueVolume} = 1+2+2+2 = 7}$

TENET : Performance model

➢ Latency and Bandwidth Model

- ➢ Total Volume : the total number of the tensor data accesses through the entire spacetime-stamp.
- ➢ Reuse Volume : the number of reused data across multiple spacetime-stamps.
- ➢ Unique Volume : the number of unique tensor data that are accessed.
- ➢ Spatial Reuse Volume : the amount of data reuse across multiple space-stamps.
- ➢ Temporal Reuse Volume : the amount of data reuse across multiple time-stamps within the same PE.

$$Delay_{read} = \frac{UniqueVolume(Input)}{bandwidth}$$

$$Delay_{write} = \frac{UniqueVolume(Output)}{bandwidth}$$

Sum of loop instances

$$Delay_{compute} = \frac{sum(D_S)}{Util_{PE} \times PE\ size}$$

$$IBW = \frac{SpatialReuseVolume}{Delay_{compute}}$$

Interconnect bandwidth

$$SBW = \frac{UniqueVolume}{Delay_{compute}}$$

Scratchpad bandwidth

TENET : Evaluation

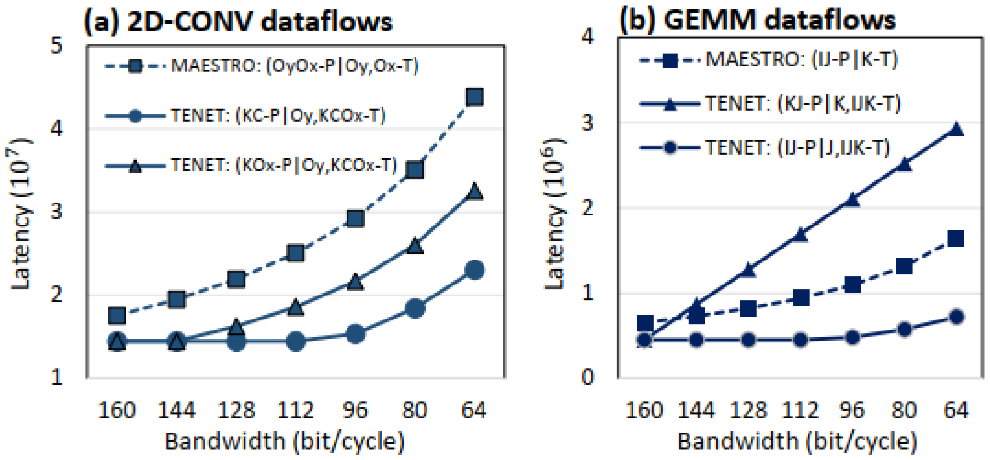
➢ Dataflow comparison

Benchmark	Dataflow	Relation-centric	Data-centric (Tp:Temporal, Sp:Spatial)
GEMM	(IJ-P ∣ J,IJK-T) applied in TPU [22]	{S[i,j,k] → PE[i%8,j%8]} {S[i,j,k] → T[fl(i/8),fl(j/8),i%8+j%8+k]}	×
	(KJ-P ∣ K,IJK-T)	{S[i,j,k] → PE[k%8,j%8]} {S[i,j,k] → T[fl(j/8),fl(k/8),i+j%8+k%8]}	×
	(IK-P ∣ K,IJK-T)	{S[i,j,k] → PE[i%8,k%8]} {S[i,j,k] → T[fl(i/8),fl(k/8),j+i%8+k%8]}	×
	(K-P ∣ I,J-T)	{S[i,j,k] → PE[k%64]} {S[i,j,k] → T[fl(k/64),i,j]}	1. SpMap(1,1) K 2. TpMap(1,1) I 3. TpMap(1,1) J
	(J-P ∣ I,K-T)	{S[i,j,k] → PE[j%64]} {S[i,j,k] → T[fl(j/64),i,k]}	1. SpMap(1,1) J 2. TpMap(1,1) I 3. TpMap(1,1) K
2D-CONV	(KC-P ∣ O_Y,KCO_X-T)	{S[k,c,ox,oy,rx,ry] → PE[k%8,c%8]} {S[k,c,ox,oy,rx,ry] → T[fl(k/8),fl(c/8),oy, k%8+c%8+ox]}	×
	(KO_X-P ∣ O_Y,KO_XC-T)	{S[k,c,ox,oy,rx,ry] → PE[k%8,ox%8]} {S[k,c,ox,oy,rx,ry] → T[fl(k/8),fl(ox/8),oy,k%8+ox%8+c]}	×
	(KC-P ∣ C,KO_X-T)	{S[k,c,ox,oy,rx,ry] → PE[k%8,c%8]} {S[k,c,ox,oy,rx,ry] → T[oy, fl(c/8),k%8+ox]}	×
	(K-P ∣ O_X,O_Y-T)	{S[k,c,ox,oy,rx,ry] → PE[k%64]} {S[k,c,ox,oy,rx,ry] → T[fl(k/64),c,ox,oy]}	1. SpMap(1,1) K; 2. TpMap(1,1) C; 3. TpMap(Sz(R_X),1) X; 4. TpMap(Sz(R_Y),1) Y; 5. TpMap(Sz(R_Y),Sz(R_Y)) R_Y; 6. TpMap(Sz(R_X),Sz(R_X)) R_X;
	(C-P ∣ O_Y,O_X-T)	{S[k,c,ox,oy,rx,ry] → PE[c%64]} {S[k,c,ox,oy,rx,ry] → T[fl(c/64),k,oy,ox]}	1. SpMap(1,1) C; 2. TpMap(1,1) K; 3. TpMap(Sz(R_Y),1) Y; 4. TpMap(Sz(R_X),1) X; 5. TpMap(Sz(R_Y),Sz(R_Y)) R_Y; 6. TpMap(Sz(R_X),Sz(R_X)) R_X;
	($R_Y O_Y$-P ∣ O_Y,O_X-T) Motivated by Eyeriss[10]	{S[k,c,ox,oy,rx,ry] → PE[ry+3*(c%4),oy]} {S[k,c,ox,oy,rx,ry] → T[fl(k/16),fl(c/16),ox]}	1. TpMap(4,4) C; 2. TpMap(16,16) K; 3. SpMap(Sz(R_Y),1) Y; 4. TpMap(Sz(R_X),1) X; 5. Cluster(Sz(R_Y),P); 6. TpMap(1,1) C; 7. TpMap(1,1) K; 8. SpMap(1,1) Y; 9. SpMap(1,1) R_Y;
	($O_Y O_X$-P ∣ O_Y,O_X-T) Motivated by Shi-diannao[15]	{S[k,c,ox,oy,rx,ry] → PE[oy%8,ox%8]} {S[k,c,ox,oy,rx,ry] → T[k,c,fl(oy/8),fl(ox/8)]}	1. TpMap(1,1) K; 2. TpMap(1,1) C; 3. SpMap(Sz(R_Y), 1) Y; 4. TpMap(10,8) X; 5. TpMap(Sz(R_Y), Sz(R_Y)) R_Y; 6. TpMap(Sz(R_X), Sz(R_X)) R_X; 7. Cluster(8, P); 8. SpMap(Sz(R_X), 1) X;
	(KC-P ∣ O_Y,O_X-T) Motivated by NVIDIA[38]	{S[k,c,ox,oy,rx,ry] → PE[k%8,c%8]} {S[k,c,ox,oy,rx,ry] → T[fl(k/8),fl(c/8),oy,ox]}	1. SpMap(1,1) K; 2. TpMap(8,8) C; 3. TpMap(Sz(R_Y),Sz(R_Y)) R_Y; 4. TpMap(Sz(R_X),Sz(R_X)) R_X; 5. TpMap(Sz(R_Y),1) Y; 6. TpMap(Sz(R_X),1) X; 7. Cluster(8, P); 8. SpMap(1,1) C;

TENET : Evaluation

➢ Dataflow comparison

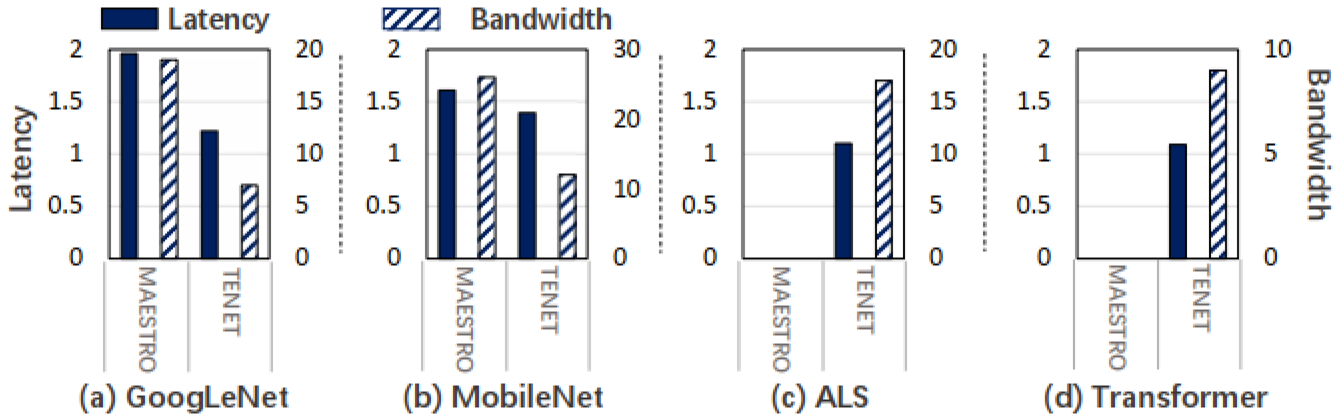
- ➢ Two dataflows (KC − P | OY,KCOX − T) (KOX − P | OY,KCOX − T) cannot be represented in data-centric notation as they require affine transformation, which means relation-centric notation is more expressive than data-centric notation.



- ➢ As a result, relation-centric notation opens up more opportunities for a larger exploration of dataflows, which is essential for designing efficient spatial architectures.
- ➢ Overall, TENET achieves 37.4% and 51.4% latency improvement on average compared with the data-centric notation by identifying more sophisticated dataflows for 2DCONV and GEMM, respectively.

TENET : Evaluation

➢ Dataflow comparison



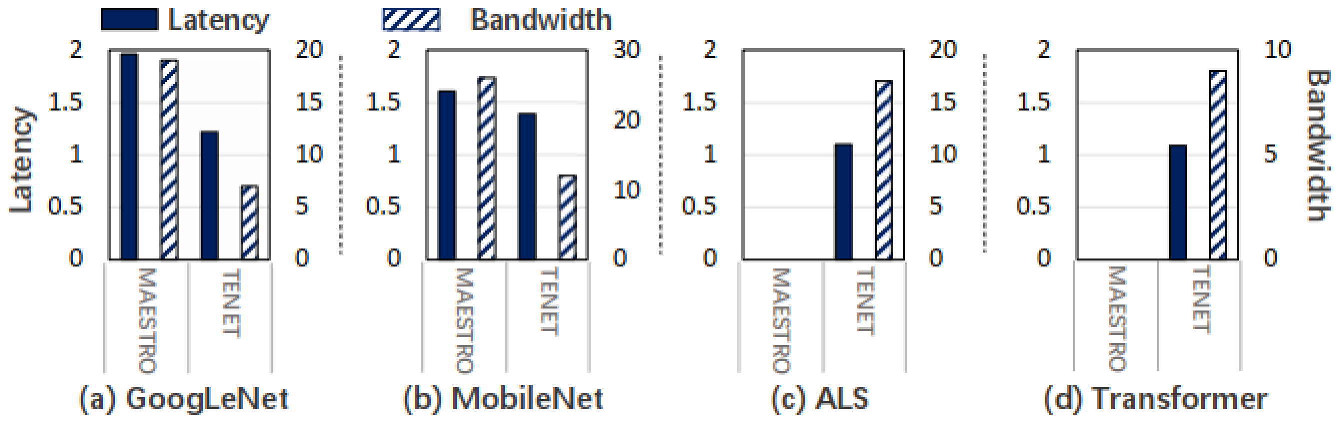
➢ The latency and bandwidth requirement of the optimal MAESTRO dataflow and TENET dataflow.

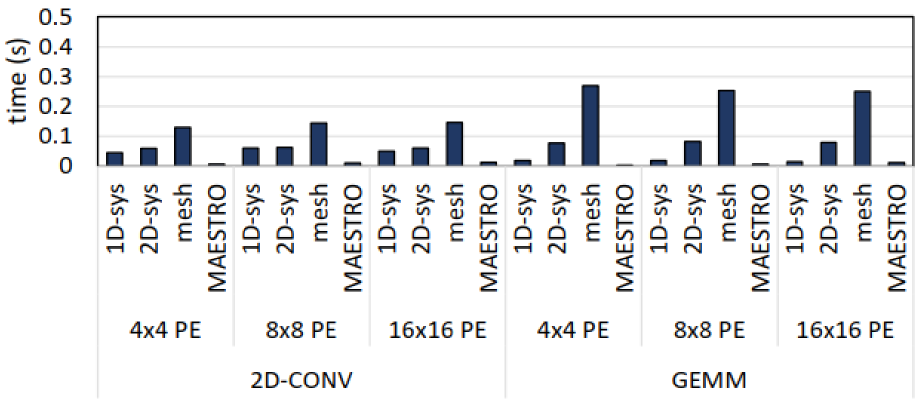
➢ The latency is normalized to the ideal latency with theoretical performance (calculated as: # of multipliers × frequency).

➢ Overall, TENET shows 74% and 22% latency reduction, and reduces the bandwidth requirement by 63% and 54% for GoogleNet and MobileNet, respectively.

TENET : Evaluation

➢ Dataflow comparison

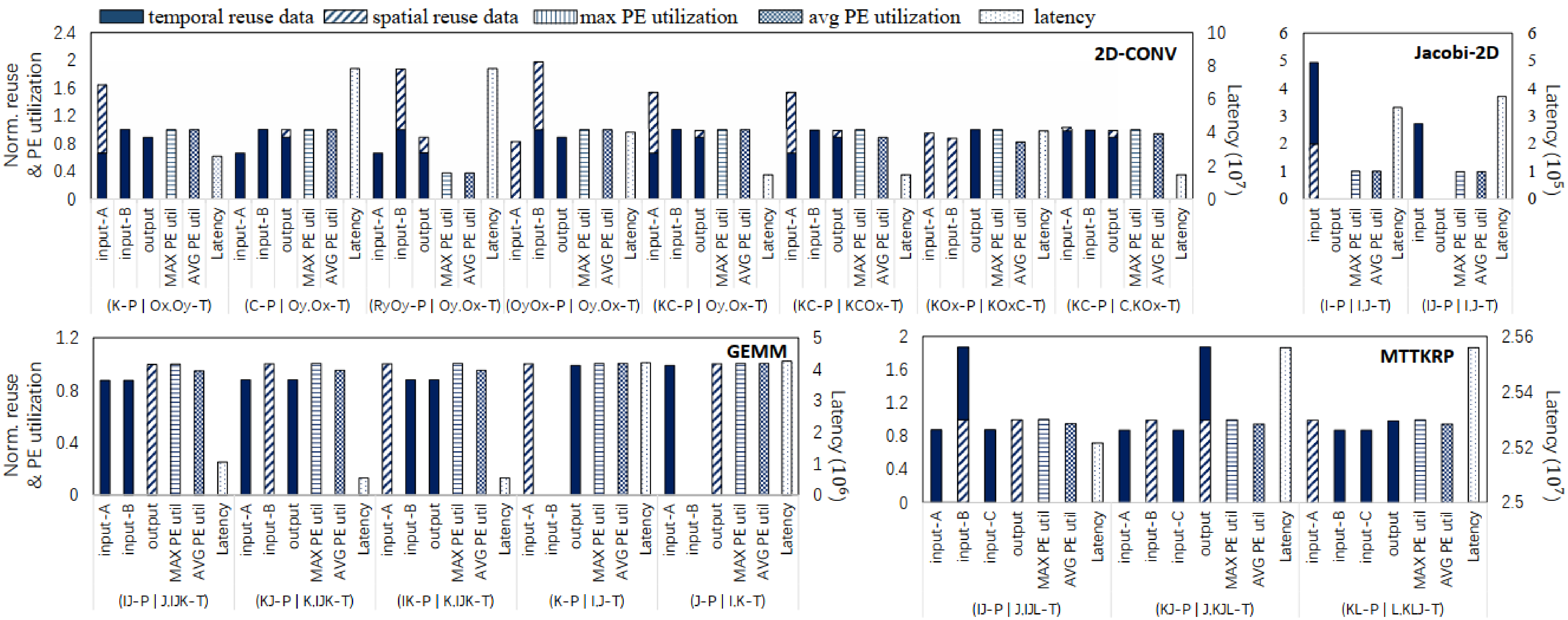




- ➢ The latency and bandwidth requirement of the optimal MAESTRO dataflow and TENET dataflow.
- ➢ The latency is normalized to the ideal latency with theoretical performance (calculated as: # of multipliers × frequency).
- ➢ Overall, TENET shows 74% and 22% latency reduction, and reduces the bandwidth requirement by 63% and 54% for GoogleNet and MobileNet, respectively.
- ➢ On average, the modeling time of a single dataflow is 10−2 second for MAESTRO, and 10−1 second for TENET.
  - ➢ The difference mainly comes from the fact that TENET models the dataflow as an integer linear programming problem, and considers more architectural details in the evaluation (e.g., interconnection, data assignment).

TENET : Evaluation

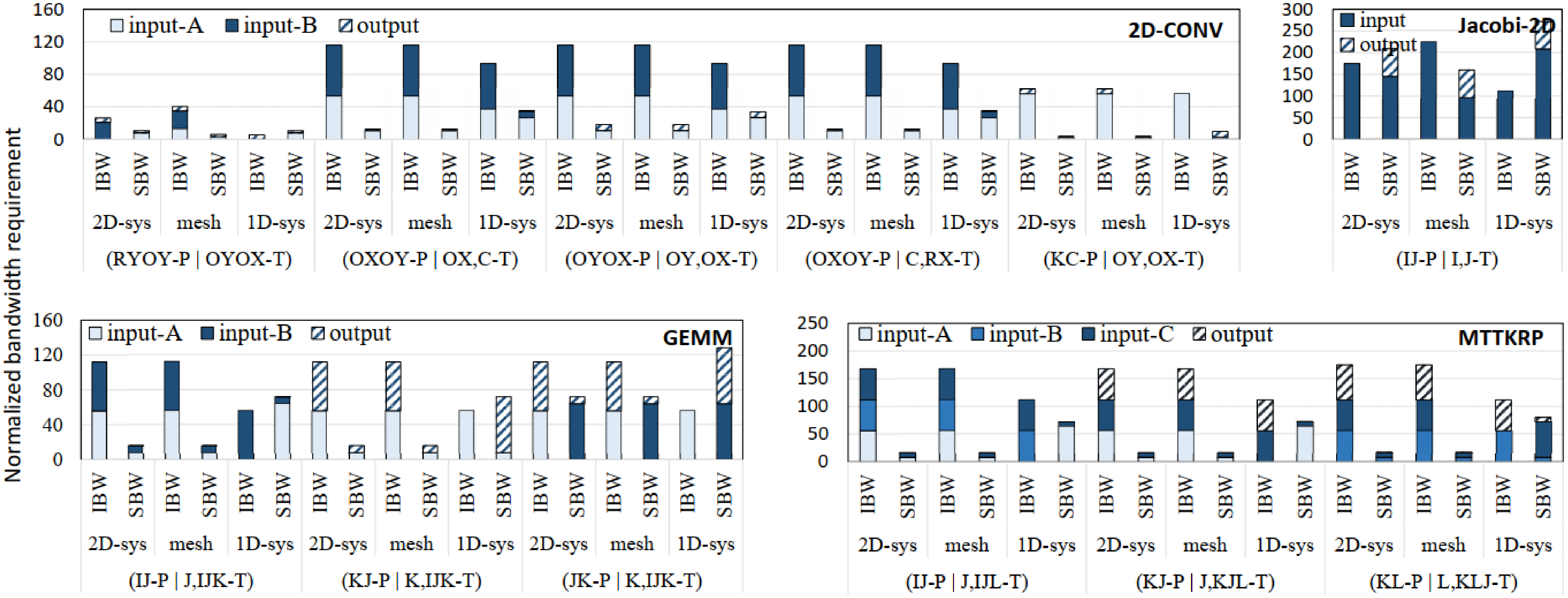
➢ Performance Metrics Evaluation



Noting that a good dataflow needs to provide both high PE utilization and data reuse. Dataflows that have poor performance usually fail in one of these two aspects. The results also demonstrate that our framework is capable of capturing spatial and temporal reuse separately.

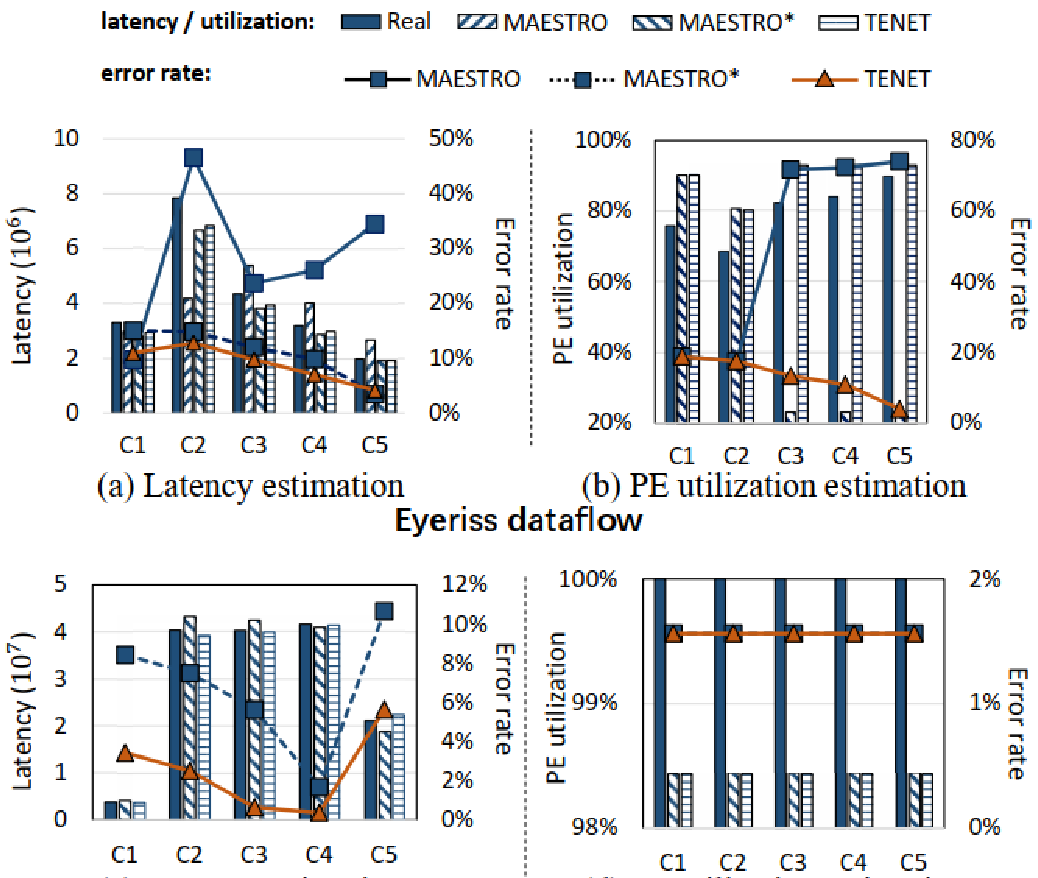
TENET : Evaluation

➢ Performance Metrics Evaluation



TENET : Evaluation

➢ Performance Metrics Evaluation



(a) Latency estimation (b) PE utilization estimation

Eyeriss dataflow

(c) Latency estimation (d) PE utilization estimation

MAERI dataflow

More accurate estimation on both latency, PE utilization

Q & A