ViDA: Video Diffusion Transformer Acceleration with Differential Approximation and Adaptive Dataflow

Li Ding[1], Jun Liu[12], Shan Huang[1], Guohao Dai[12]*

[1]*Shanghai Jiao Tong University* [2]*Infinigence AI*

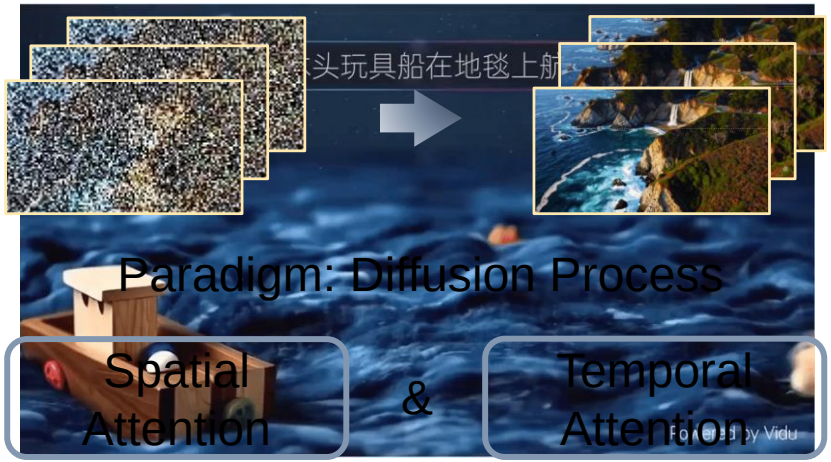
Equal Contribution *Corresponding to:<daiguohao@sjtu.edu.cn>

ASP-DAC 2025

2025.1.21

Background

- ***Current video generation models excel due to the diffusion paradigm and the Video Diffusion Transformer backbone.***



Backbone: Video Diffusion Transformer

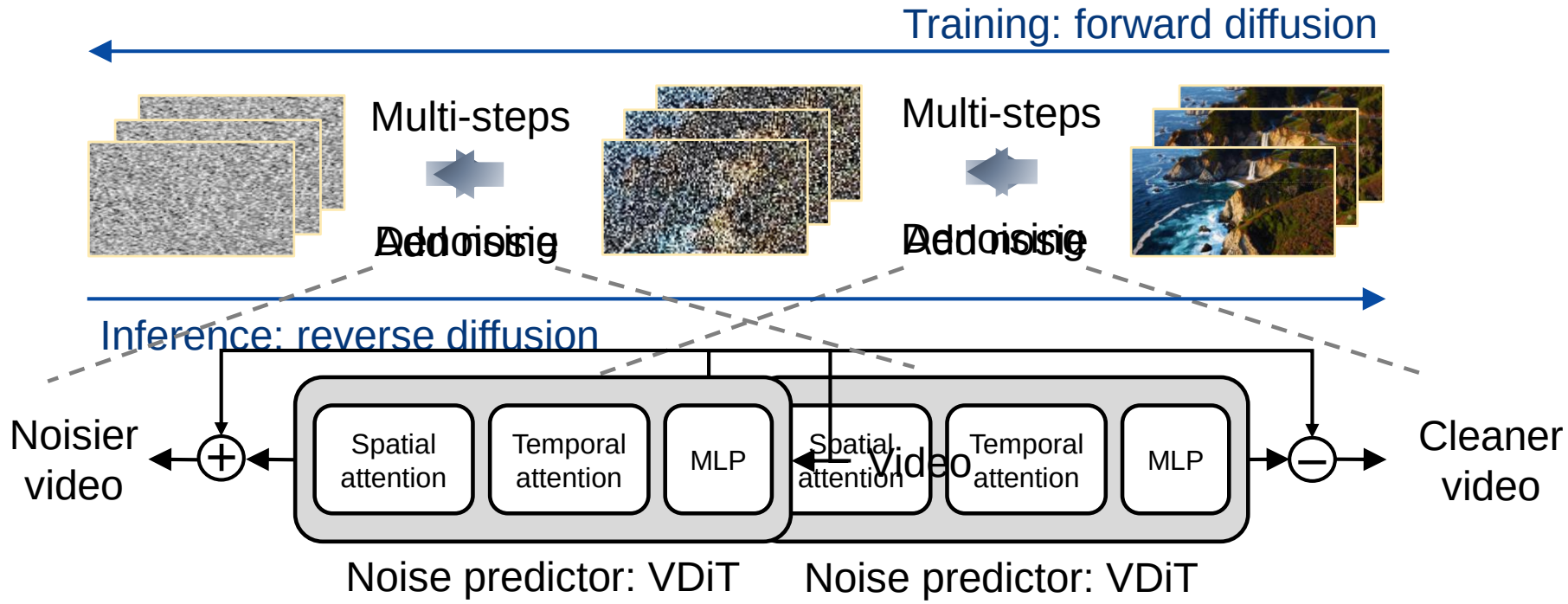
Vidu [1] : Tokyo Train

Sora [2] : Clifftop Waves and Waterfalls

[1] https://www.vidu.studio/ [2] https://sora.com

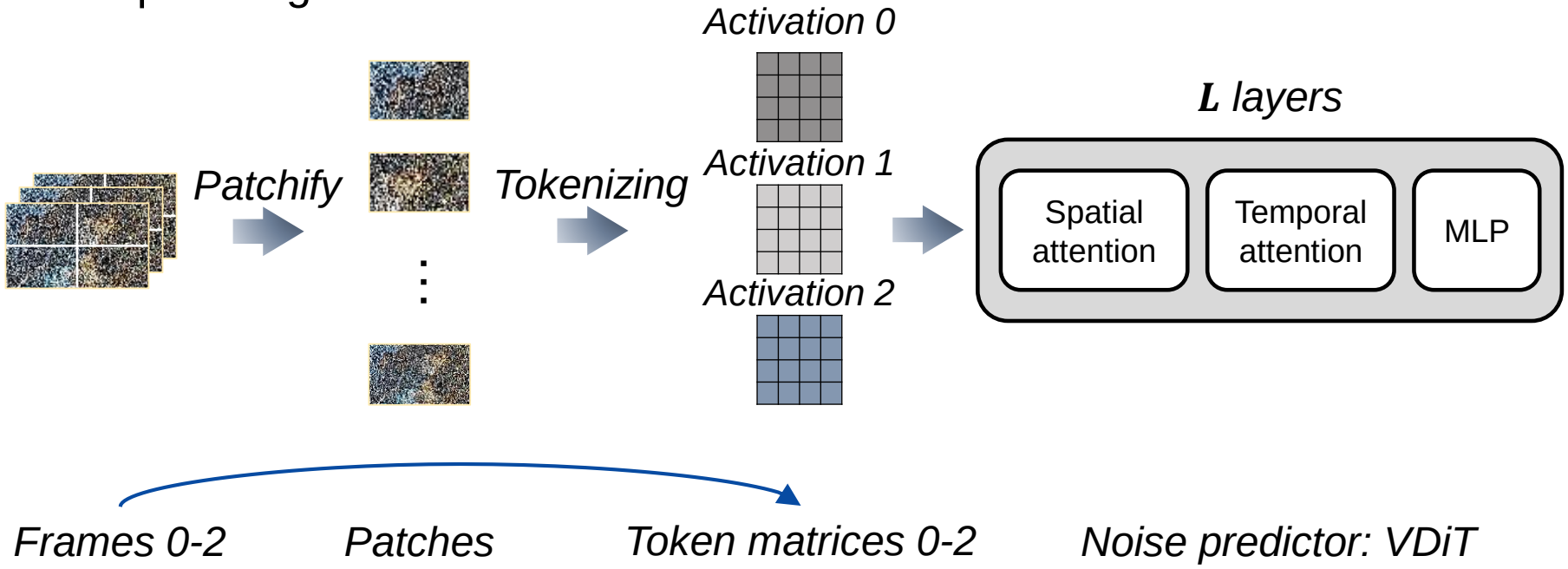
Background

- ***The Paradigm: What is diffusion model ?***
  - Frames are iteratively denoised, with noise predicted by VDiT.



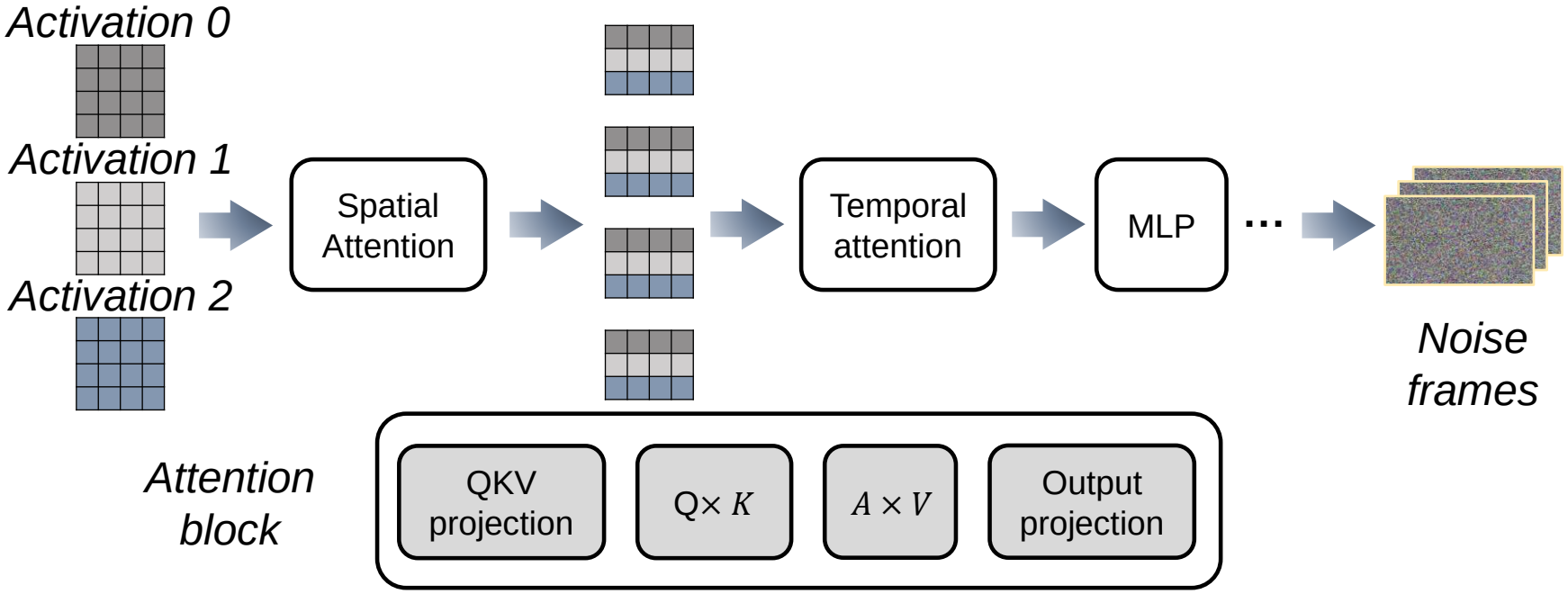
Background

- ***The Backbone: What is Video Diffusion Transformer (VDiT) ?***
  - A transformer-based noise predictor with a spatial and temporal patching structure.



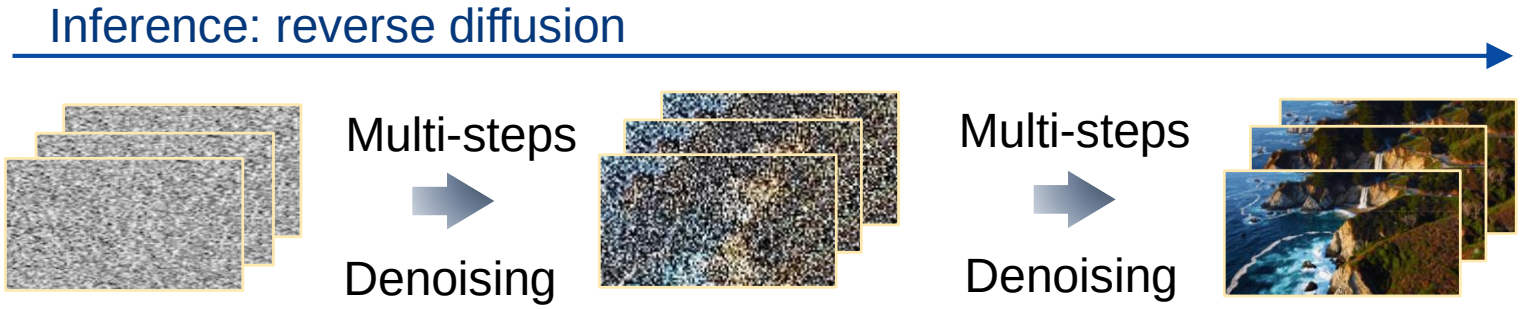
Background

- ***The Backbone: What is Video Diffusion Transformer (VDiT) ?***
  - A transformer-based noise predictor with a spatial and temporal patching structure.



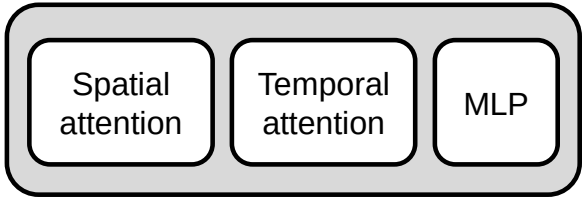
Background

- ***Video generation is slow due to large inference data.***
  - It costs over 1 hour to generate a one-minute video.[1]



VDiT is the main bottleneck!
Take up over 90% of the inference time.

*Other processes include initialization and noise handling.



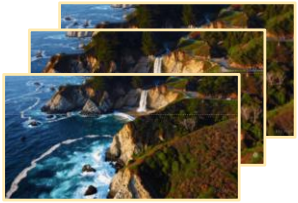
Noise predictor: VDiT

[1] https://www.wired.com/story/openai-sora-generative-ai-video/

Key Idea

- ***How to accelerate VDiT ?***
  - We can learn from traditional **video processing.**

Video Processing[1]



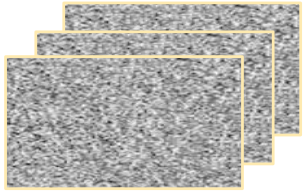
Original video

Patchify → Intra-frame prediction → Inter-frame prediction

Compressed video

Lack of compression process!

Video Generation



Noisy video

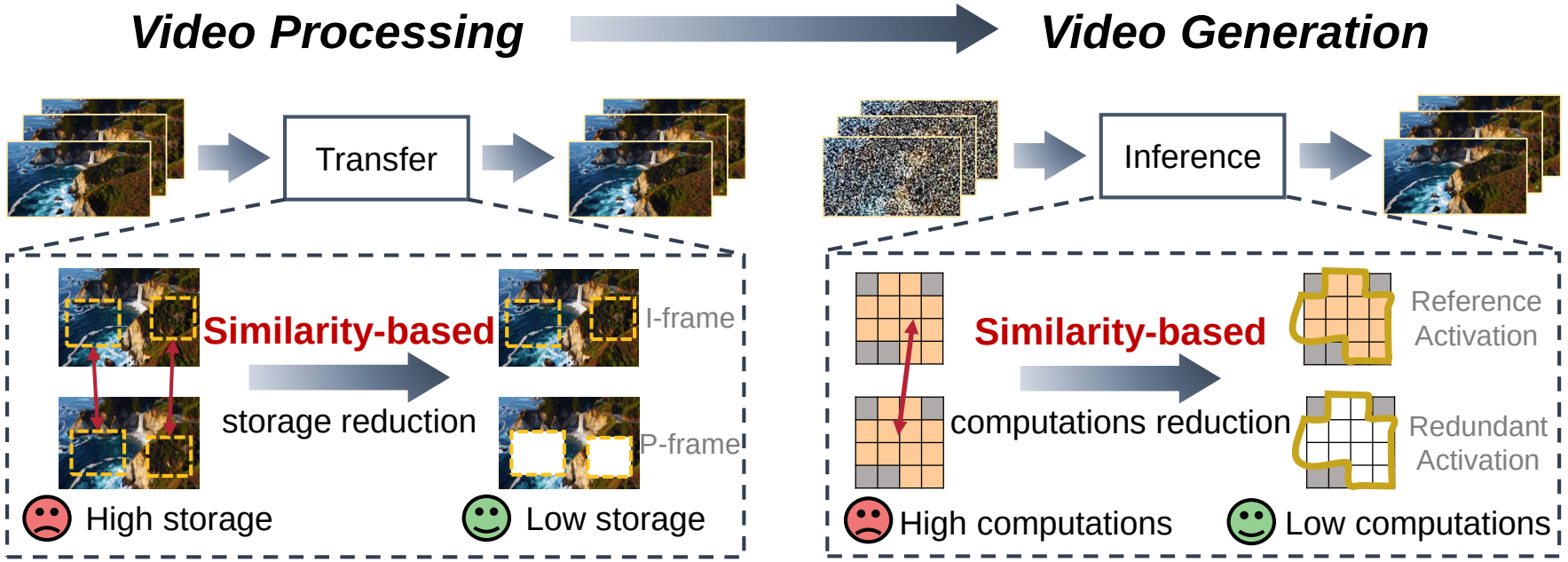
Patchify → Spatial attention → Temporal attention

Clean video

[1] Research on Key Technologies of H.264/HEVC Video Coding & Decoding Standard.

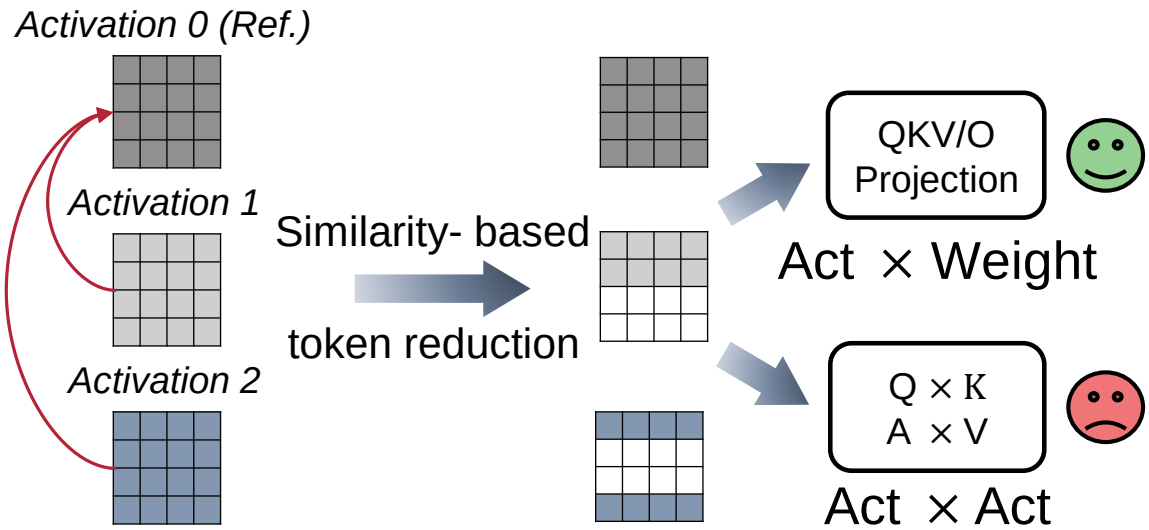
Key Idea

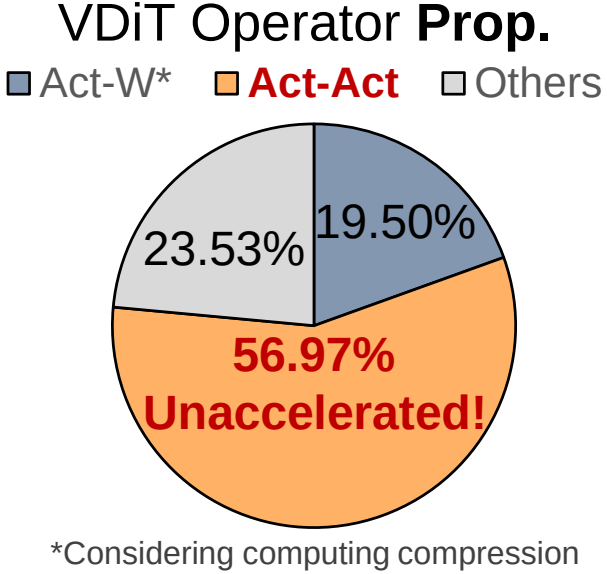
- ***How to accelerate VDiT ?***
  - We can learn from traditional **video processing.**



Challenge 1

- ***VDiT exists large unaccelerated computations.***
  - Existing token reduction methods[1][2] ***fail to accelerate Act-Act operator*** due to accuracy loss caused by dimension mismatch.





*Considering computing compression

[1] Song Z, et al. CMC: Video Transformer Acceleration via CODEC Assisted Matrix Condensing [ASPLOS 2024]
[2] Wang X, et al. InterArch: Video Transformer Acceleration via Inter-Feature Deduplication with Cube-based Dataflow [DAC 2024]

Contribution 1

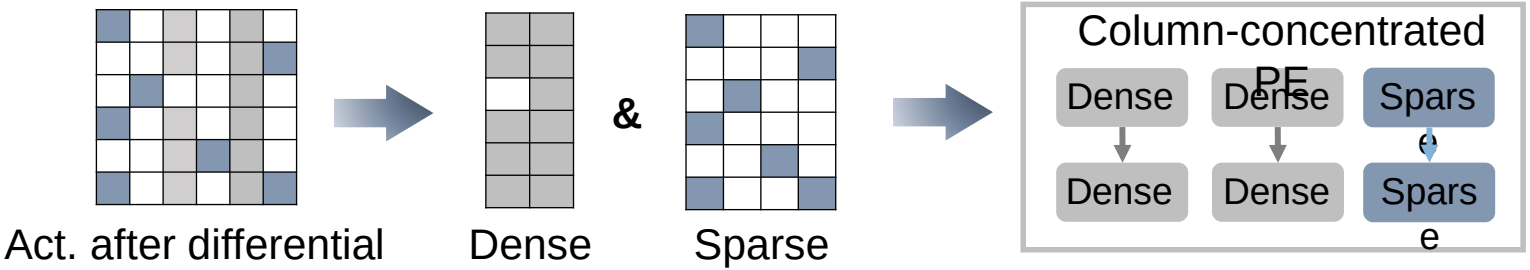
- ***Algorithm-hardware co-design for Act-Act operator acceleration.***

$$Q_2K_2^T$$ Dense computing

$$= Q_1\Delta K_2^T + \Delta Q_2 K_1^T + \Delta Q_2 \Delta K_2^T + Q_1 K_1^T$$ More computations

$$\approx Q_2\Delta K_2^T + Q_1 K_1^T$$ **Sparse computing**

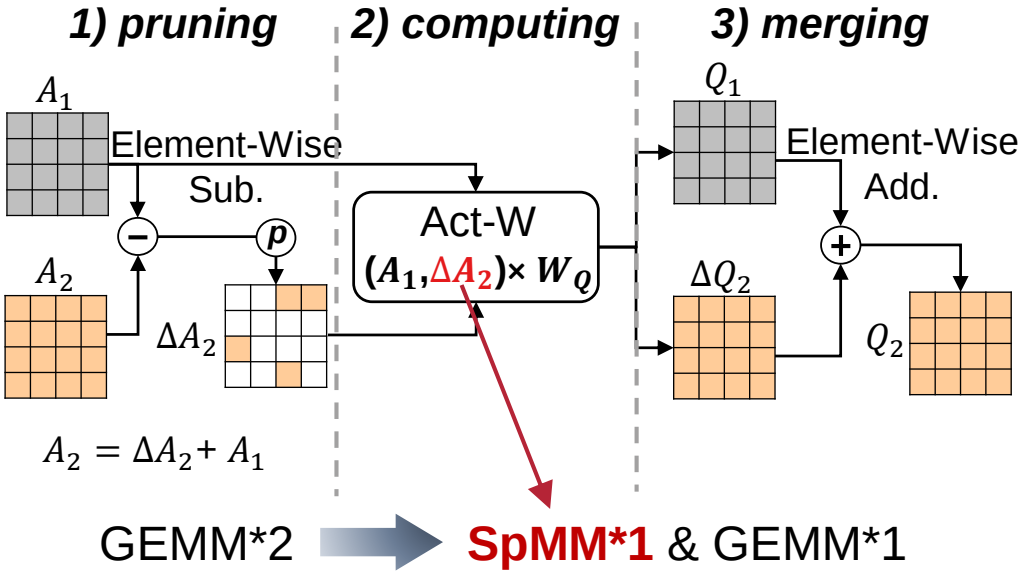
Tech 1 (algorithm): Differential Approximation Method



Tech 2 (hardware): Column-concentrated Processing Element

Tech1: Differential Approximation Method

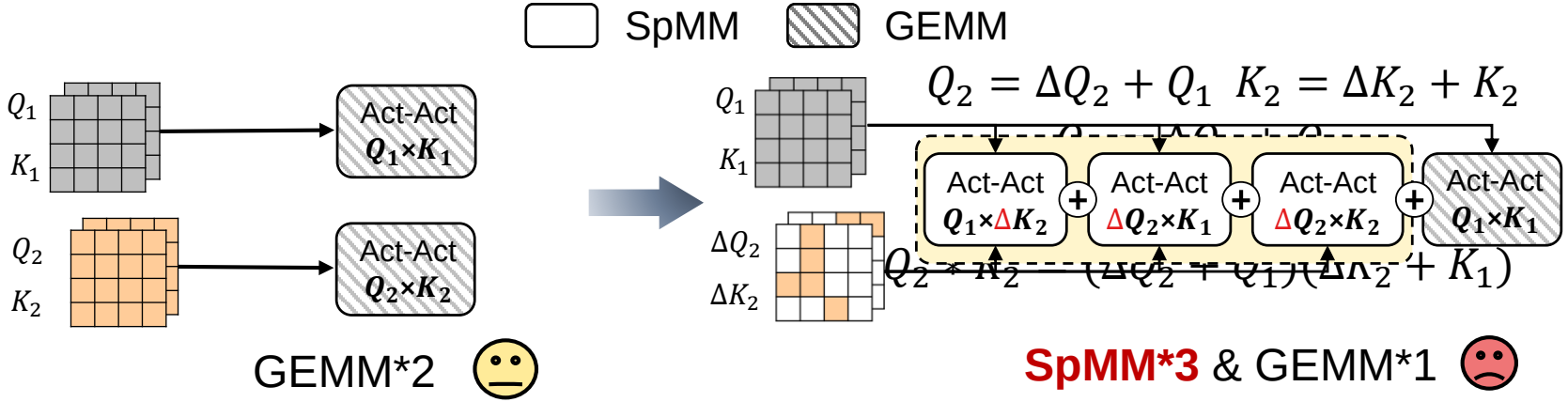
- ***What is differential computing ?***
  - Reducing redundant computations by leveraging the **similarity** between activations.



Differential computing can be applied to the **Act-W operator** efficiently.

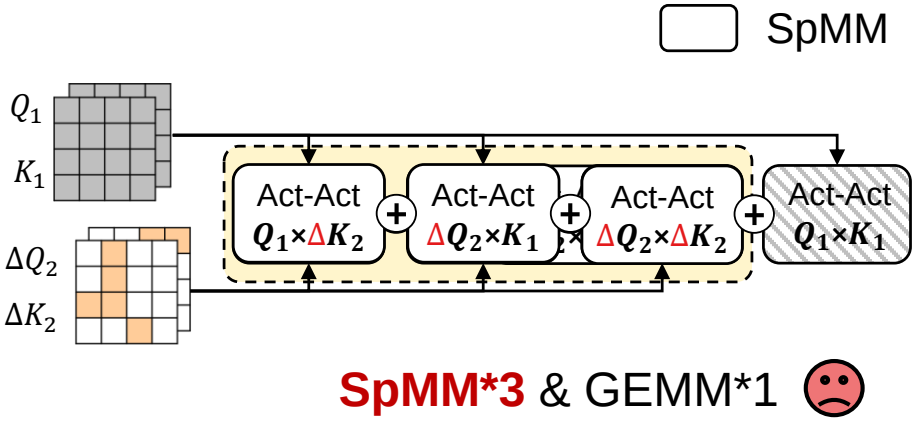
Tech1: Differential Approximation Method

- ***How can differential computing be applied to Act-Act operator ?***



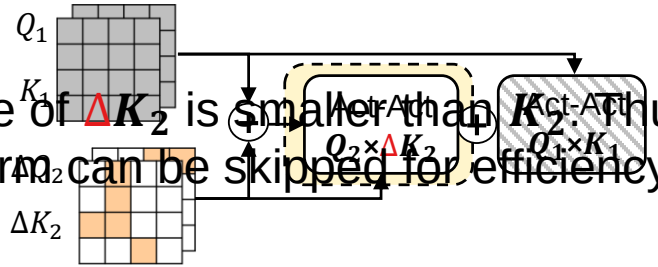
Tech1: Differential Approximation Method

- ***How can differential computing be applied to Act-Act operator ?***



SpMM*3 & GEMM*1

The value of ΔK_2 is smaller than K_2. Thus, some term can be skipped for efficiency.



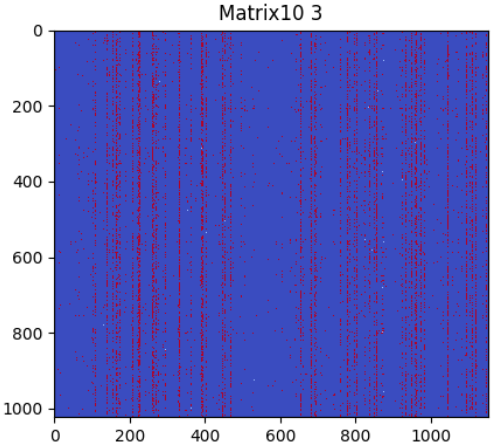
SpMM*1 & GEMM*1

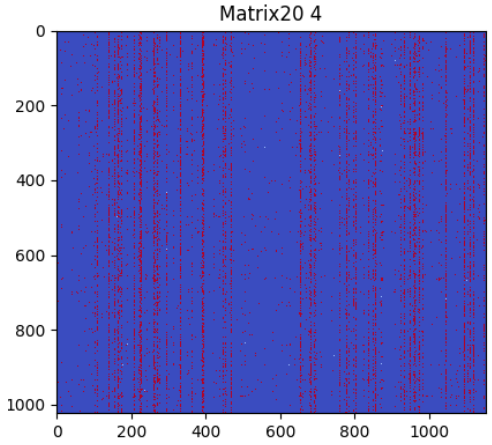
Naive differential computing → Differential approximation computing

We can reduce the computations of Act-Act operator by 51.67% with negeligible accuracy loss.

Tech2: Column-concentrated PE

- ***How to perform SpMM introduced by differential computing efficiently ?***
  - Observation: Column-split computing existing **column-sparse pattern**.



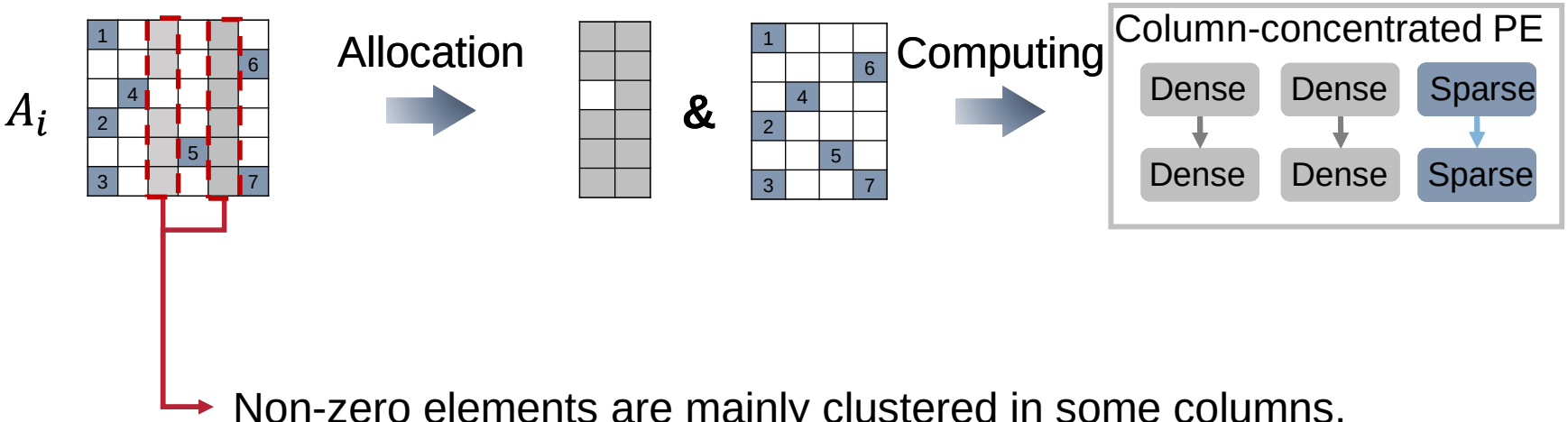


Red: Non-zero elements
Blue: zero elements

Non-zero elements are mainly clustered in some columns.

Tech2: Column-concentrated PE

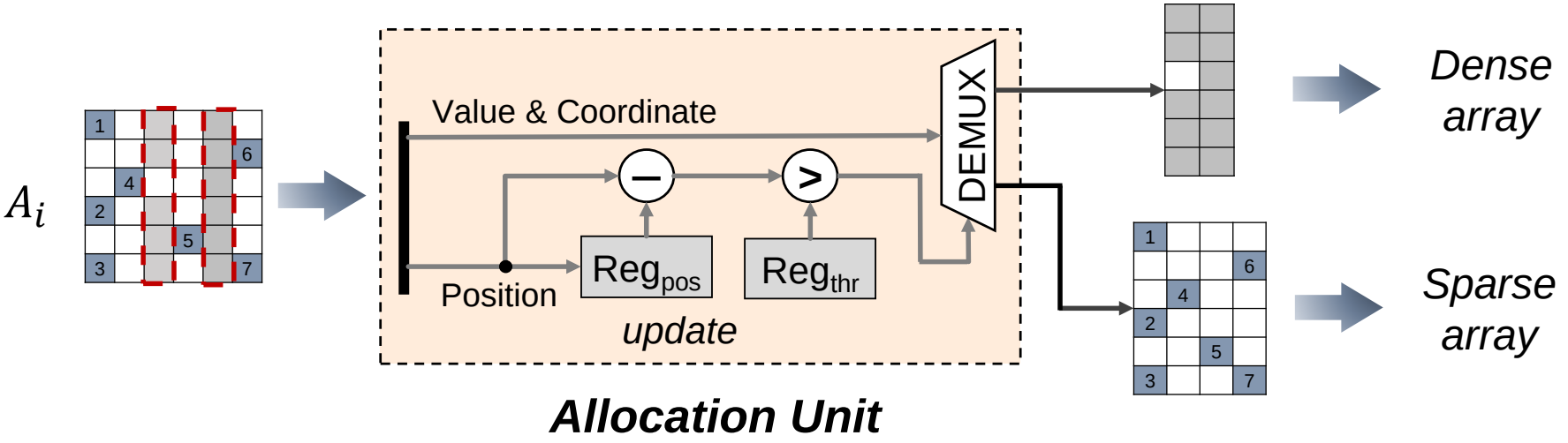
- ***How to perform SpMM introduced by differential computing efficiently ?***
  - Column-split computing based on **column-sparse pattern**.
    1) Allocation 2) Computing



Tech2: Column-concentrated PE

- ***How to perform SpMM introduced by differential computing efficiently ?***

  1) Allocation: determine dense and sparse columns by **thresholds**.

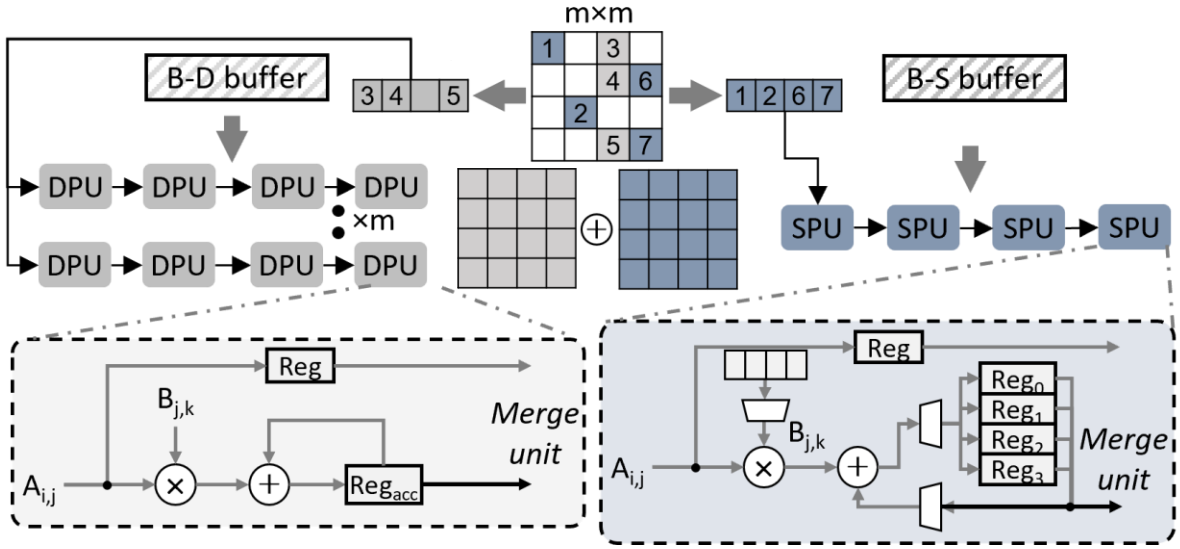


Allocation Unit

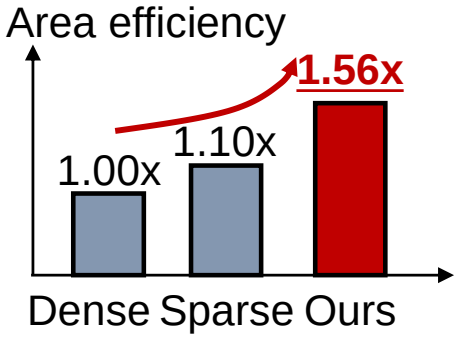
Tech2: Column-concentrated PE

- ***How to perform SpMM introduced by differential computing efficiently ?***

  2) Computing: Compute **outer products** of allocated columns using sparse and dense arrays.

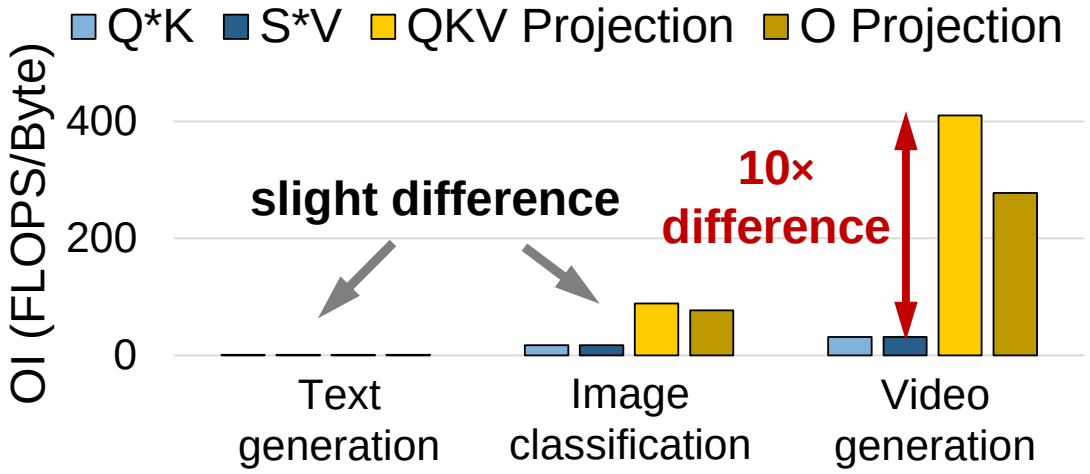


1.10x / 1.56x higher area efficiency compared with dense-only and sparse-only architectures.



Challenge 2

- ***Large operational intensity (OI) difference* among operators leads to inefficient hardware utilization.**



QKV/O Projection — *Compute-bound*

$Q \times K^T$
$A \times V$ — *Memory-bound*

<40% utilization for static architecture

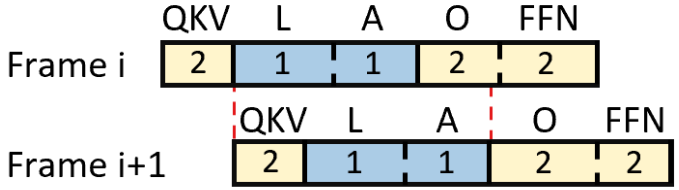
Operational intensity[1] = computations / memory accesses

[1] https://crd.lbl.gov/divisions/amcr/computer-science-amcr/par/research/roofline/

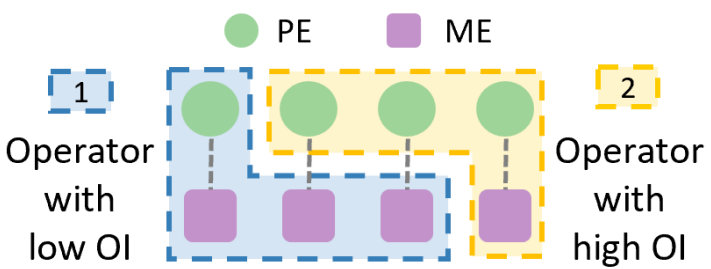
Contribution 2

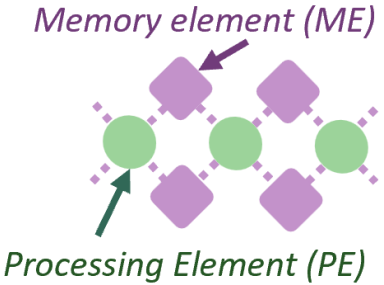
- ***Use intensity adaptive dataflow architecture design for dynamically allocate resources for different operators.***

Dataflow



Architecture

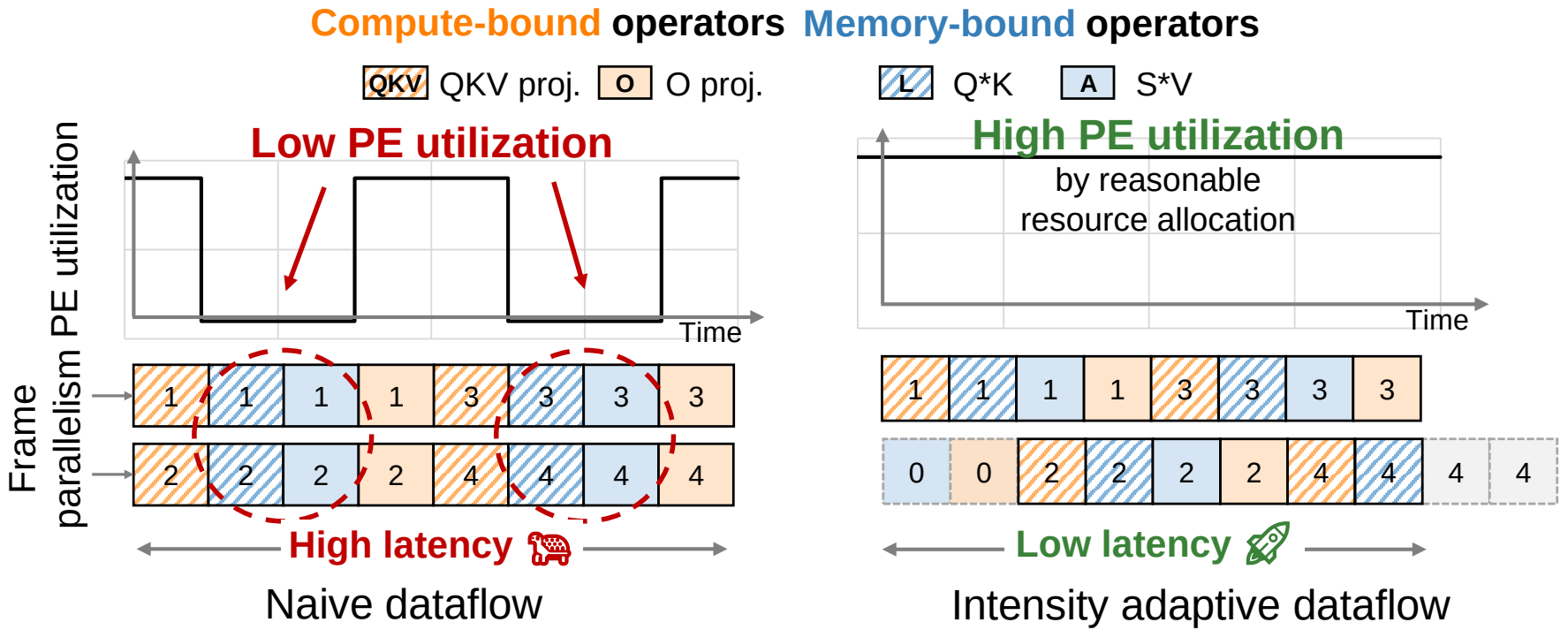




Tech 3 (dataflow): Intensity Adaptive Dataflow Architecture

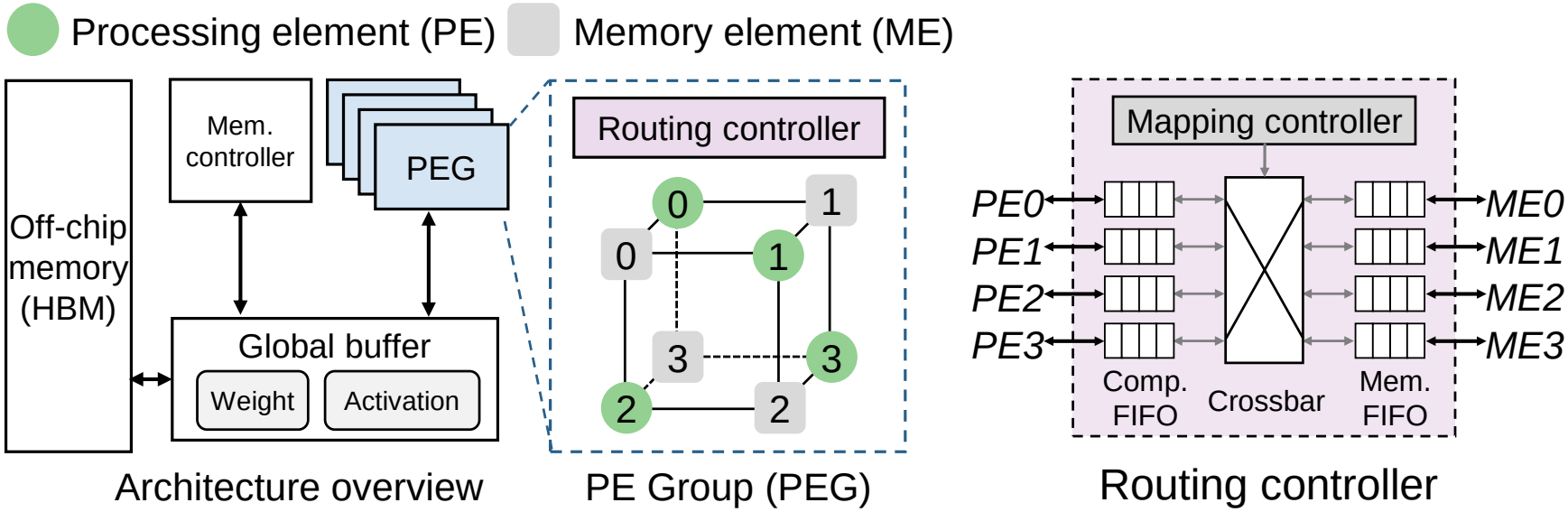
Tech3: Adaptive Dataflow Architecture

- ***Dataflow: Intensity adaptive dataflow for reasonable resource allocation.***



Tech3: Adaptive Dataflow Architecture

- ***Architecture: Reconfigurable architecture including routing controller to support flexible resource allocation.***



Evaluation

Accuracy evaluation

- ***Models:*** STDiT.1, Latte, VDT
- ***Datasets:*** UCF101, MSR-VTT
- ***Baselines***
  - NVIDIA A100 GPU
  - SOTA Vision Accelerators
    - CMC[1] [ASPLOS 2024]
    - InterArch[2] [DAC 2024]
- ***Metrics***
  - FVD, FID, CLIPSIM

	FVD↓	CLIPSIM↑	Average loss
STDiT(dense)	477.97	0.264	---
VIDA-STDiT	479.56	0.262	**0.55%**
CMC-STDiT	483.53	0.263	0.77%
Latte(dense)	505.27	0.294	---
ViDA-Latte	514.33	0.293	**1.07%**
CMC-Latte	518.59	0.293	1.45%

*We further test the full UCF-101 dataset for ViDA and the SOTA CMC.

[1] Song Z, et al. CMC: Video Transformer Acceleration via CODEC Assisted Matrix Condensing [ASPLOS 2024]
[2] Wang X, et al. InterArch: Video Transformer Acceleration via Inter-Feature Deduplication with Cube-based Dataflow [DAC 2024]

Evaluation

Area evaluation

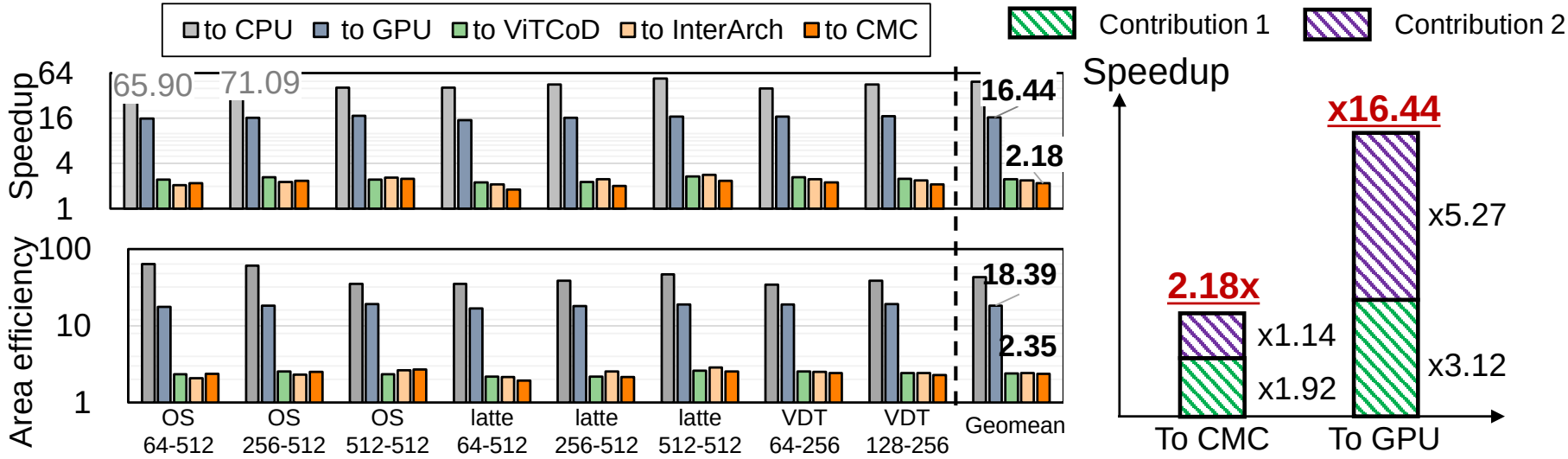
- ***Simulation process library***
  - 32nm standard librabry
- ***Tools***
  - Synopsys Design Compiler
  - CACTI 7
- ***Architecture***
  - 4 x PE group
    - 8 x column-concentrate PEs
      - 4 x 4 DPUs
      - 1 x 4 SPUs
      - Allocation unit
      - Merge unit

Component			Area (mm^2)	Breakdown
PEG × 4	CCPE × 8	4×4 DPUs	1.04	51.39%
		1×4 SPUs	0.52	25.60%
		Allocation unit	0.01	0.70%
		Merge unit	0.07	3.27%
	Routing controller		0.09	4.38%
	Memory elements × 8		0.07	3.38%
Global buffer			0.23	11.27%
Total ($7nm$)			2.03	100.00%

*Scale to $7nm$ with a frequency of 1GHz for comparing with other works

Evaluation

- ***ViDA achieves average 16.44x/2.18x speedup and 18.39x/2.35 area efficiency compared with A100 GPU/SOTA Vision Accelerator.***



You H, et al. Vitcod: Vision transformer acceleration via dedicated algorithm and accelerator co-design
Song Z, et al. CMC: Video Transformer Acceleration via CODEC Assisted Matrix Condensing [ASPLOS 2024]
Wang X, et al. InterArch: Video Transformer Acceleration via Inter-Feature Deduplication with Cube-based Dataflow [DAC 2024]

Conclusion

Thank you for your attention!

Communication e-mail : daiguohao@sjtu.edu.cn

Lab homepage

ViDA: Video Diffusion Transformer Acceleration with Differential Approximation and Adaptive Dataflow