上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

Ditto: Accelerating Diffusion Model via Temporal Value Similarity
CMC: Video Transformer Acceleration via CODEC Assisted Matrix Condensing

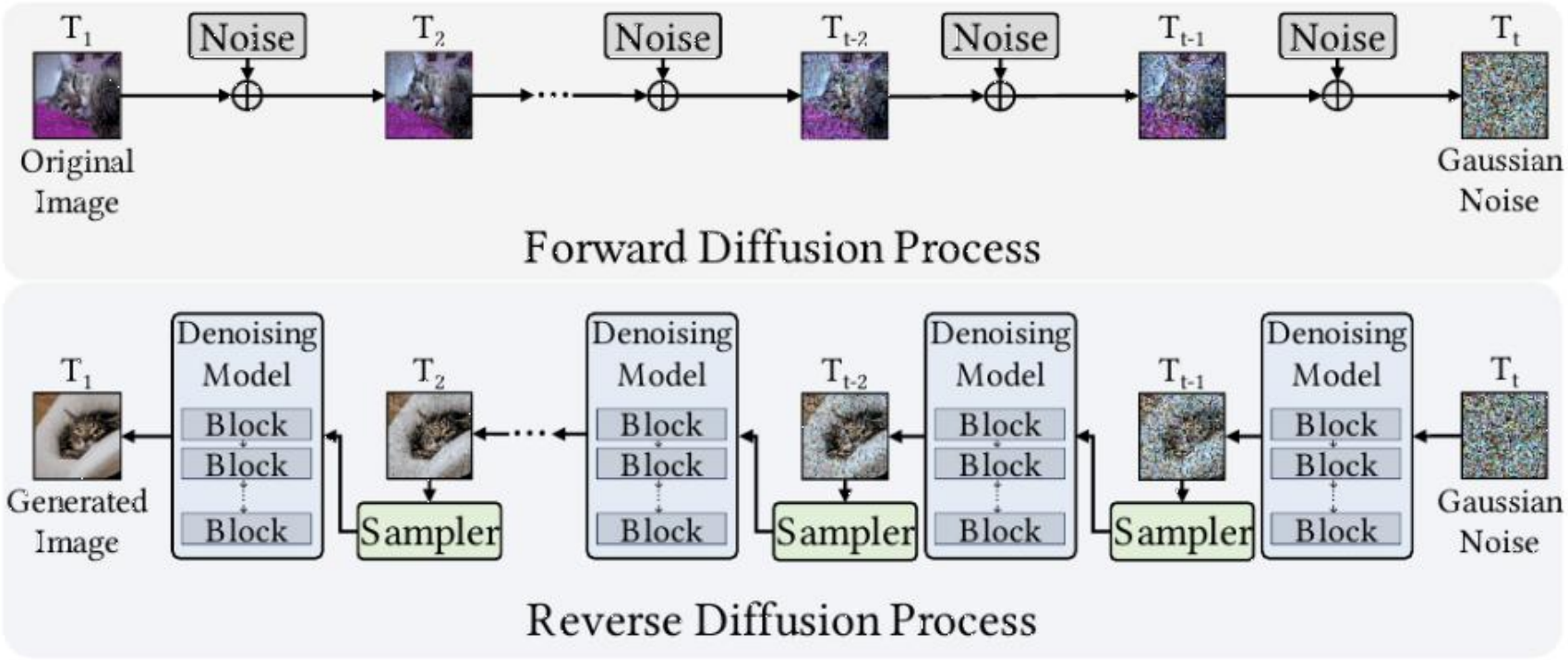
SHANGHAI JIAO TONG UNIVERSITY

刘昊淞

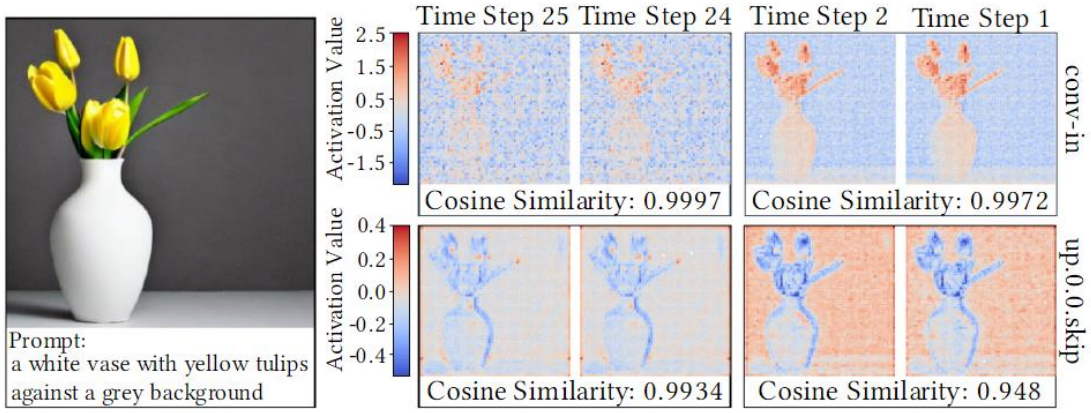
01

Ditto: Accelerating Diffusion Model via Temporal Value Similarity

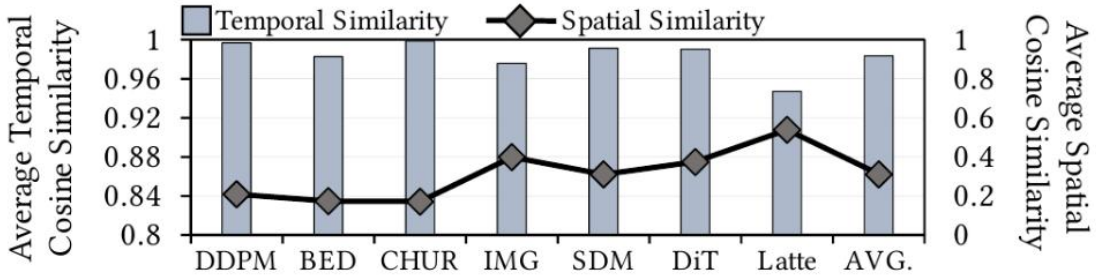
Preliminaries



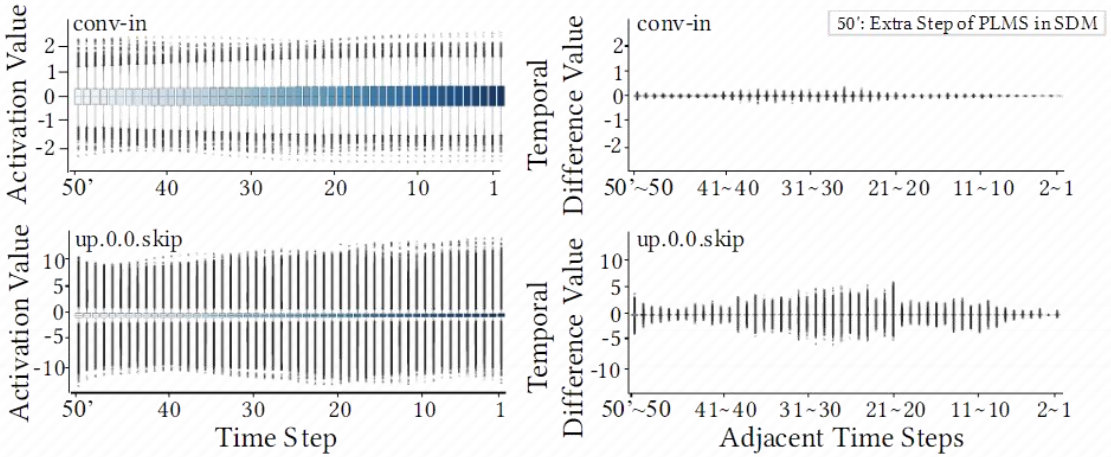
Motivation



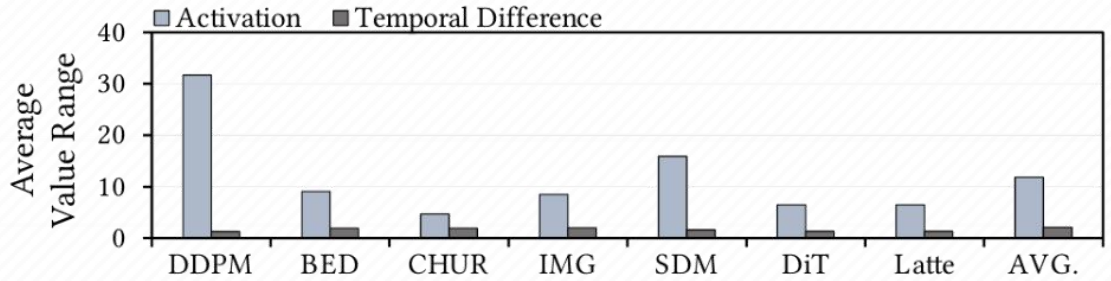
(a) Value heatmap and cosine similarity of activations between adjacent time steps in two layers of SDM [68].



(b) Average temporal similarity of activations between adjacent time steps and average spatial similarity of activations across various diffusion models.

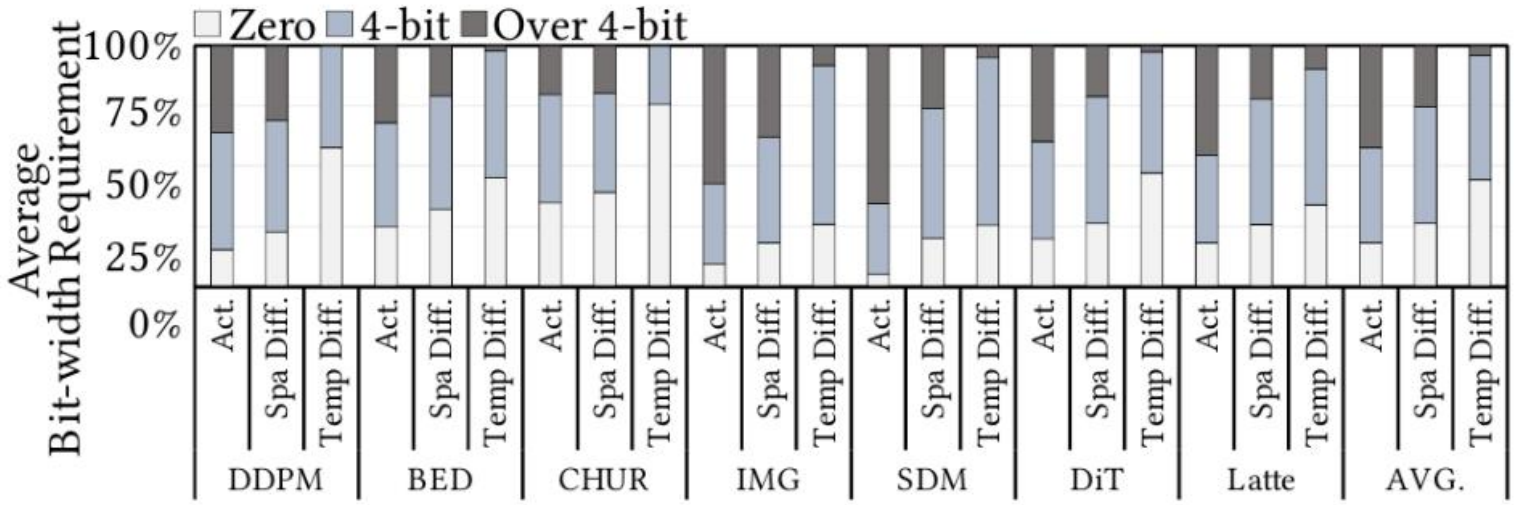


(a) Value range of activations and temporal differences across time steps in SDM [68].



(b) Average value range of activations and temporal differences in various diffusion models.

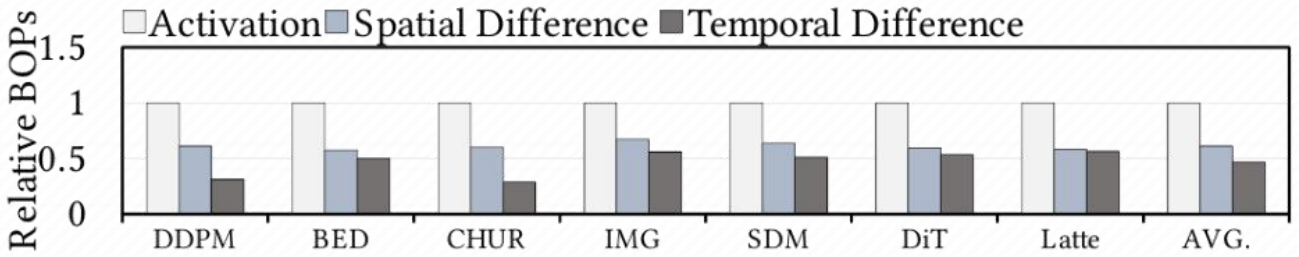
Motivation



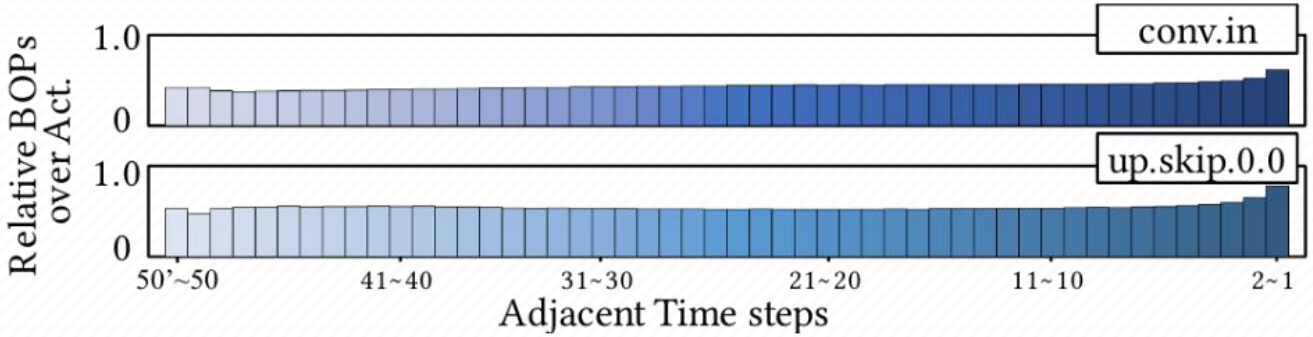
- The results show that zero temporal differences, indicating no change in values between time steps, constitute 44.48% of the total temporal differences on average.

Motivation

- The temporal difference approach can achieve 53.3% and 23.1% fewer BOPs on average compared to the original models and the spatial difference method.



(a) Relative BOPs of the various methods across diffusion models.



(b) Relative BOPs of the temporal difference approach compared to original activation across all adjacent time steps in SDM [68].

Method--Linear Layer

- Subtracting the input of the current time step from the input of the previous time steps. Detect zero differences and the differences that can be represented inlower bit-width.
- Execute the layer only with the differences in the second stage.
- Applies summation between the result of difference processing and the previous time step output, as the third stage.

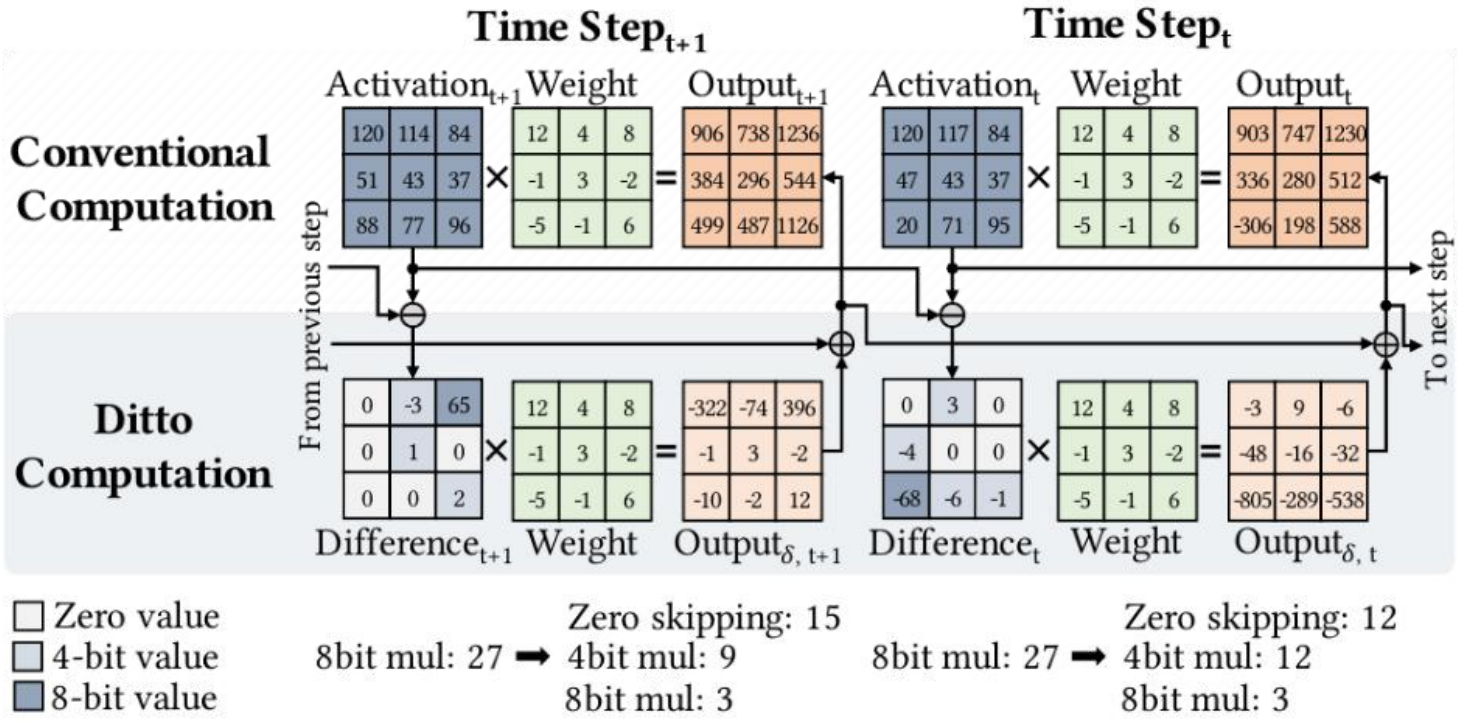


Fig. 7: Process of linear layers in the Ditto algorithm.

Method--Attention

$$
\begin{aligned}
Q_t K_t &= (Q_{t+1} + \Delta Q)(K_{t+1} + \Delta K) \\
&= Q_{t+1}K_{t+1} + \Delta Q K_{t+1} + Q_{t+1}\Delta K + \Delta Q \Delta K \\
&= Q_{t+1}K_{t+1} + \Delta Q K_{t+1} + Q_t \Delta K
\end{aligned}
$$

- The Ditto algorithm treat Qt and Kt+1 as weight. Also, the same mechanism applied to P ×V .
- In cross attention,K′ and V′ do not change with varying time steps in the layer. The Ditto algorithm treats K′ and V′ as weight.

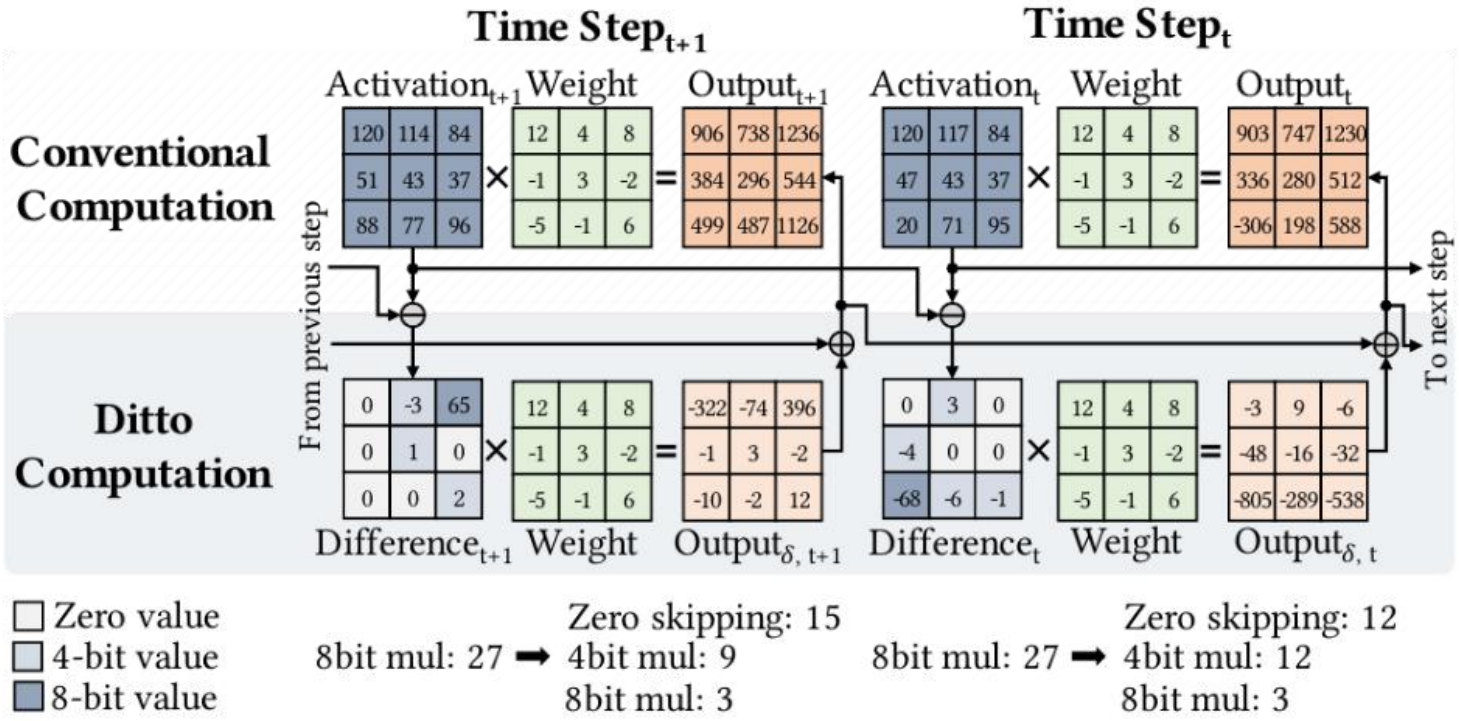
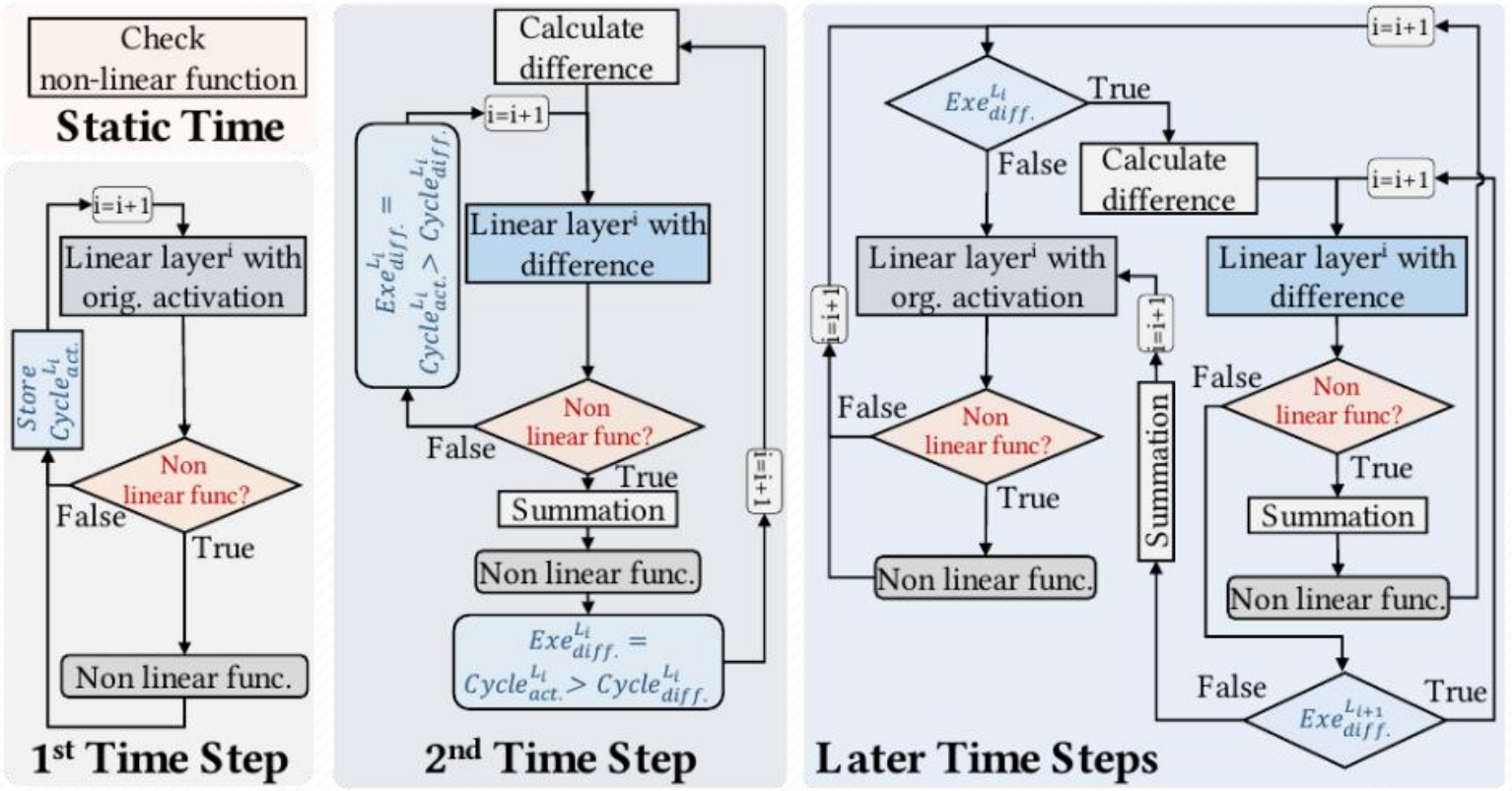


Fig. 7: Process of linear layers in the Ditto algorithm.

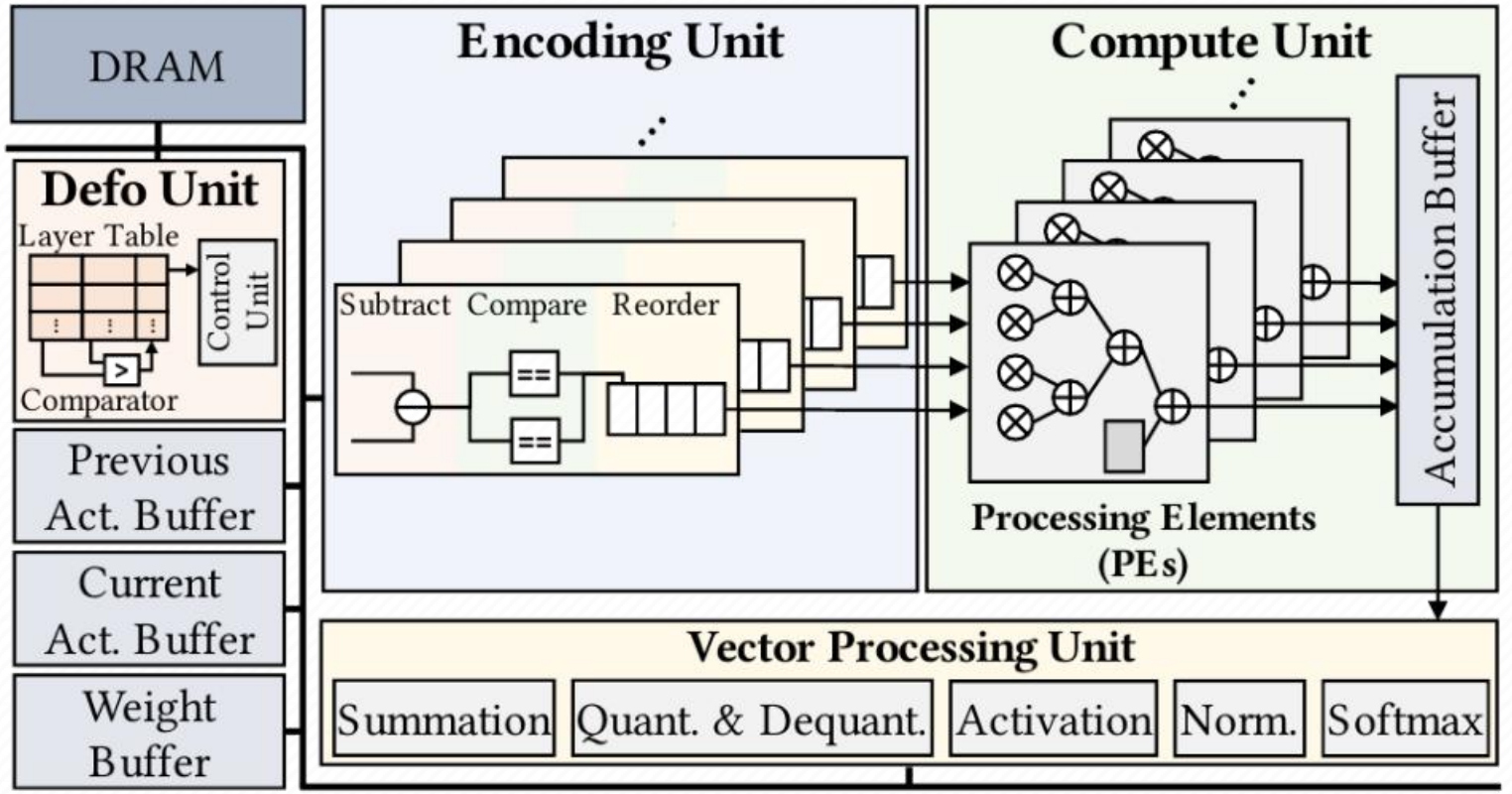
Method

- Non-linear functions require original data to ensure numerical equivalence.
- Linear layer operations require additional memory accesses to obtain the linear layer input from the previous time step in order to calculate differences.
- Therefore, some layers would be converted into memory-intensive operations due to the increased memory accesses and reduced computational intensity, even though diffusion models are compute-intensive networks



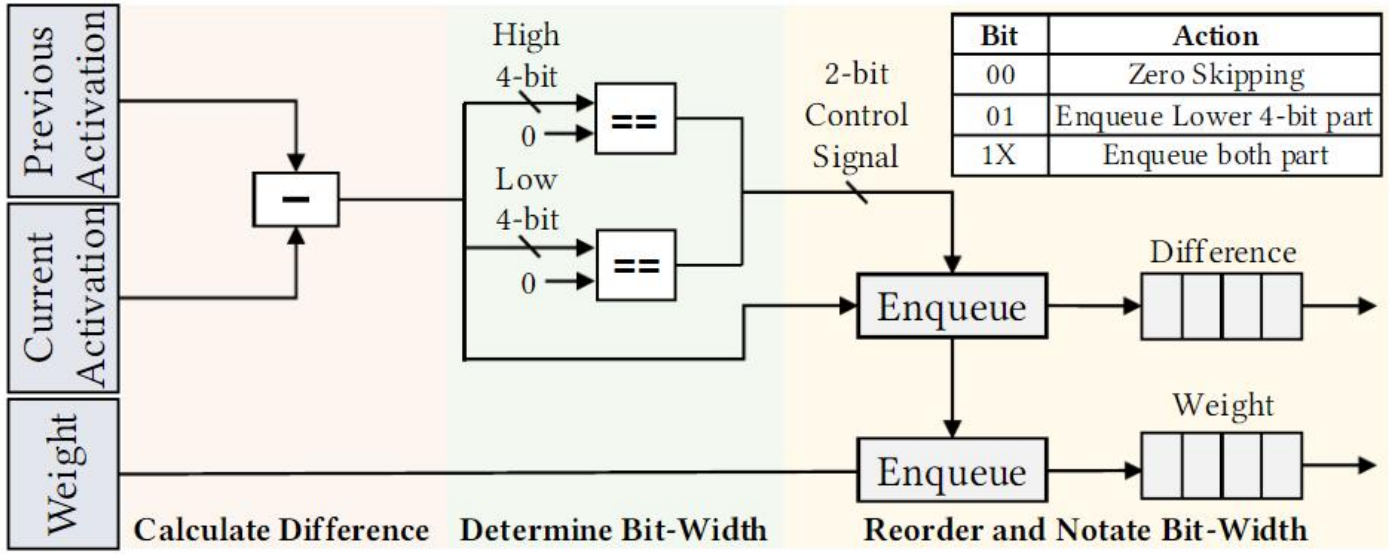
Hardware

- Encoding Unit
- Compute Unit
- Vector Processing Unit
- Defo Unit.



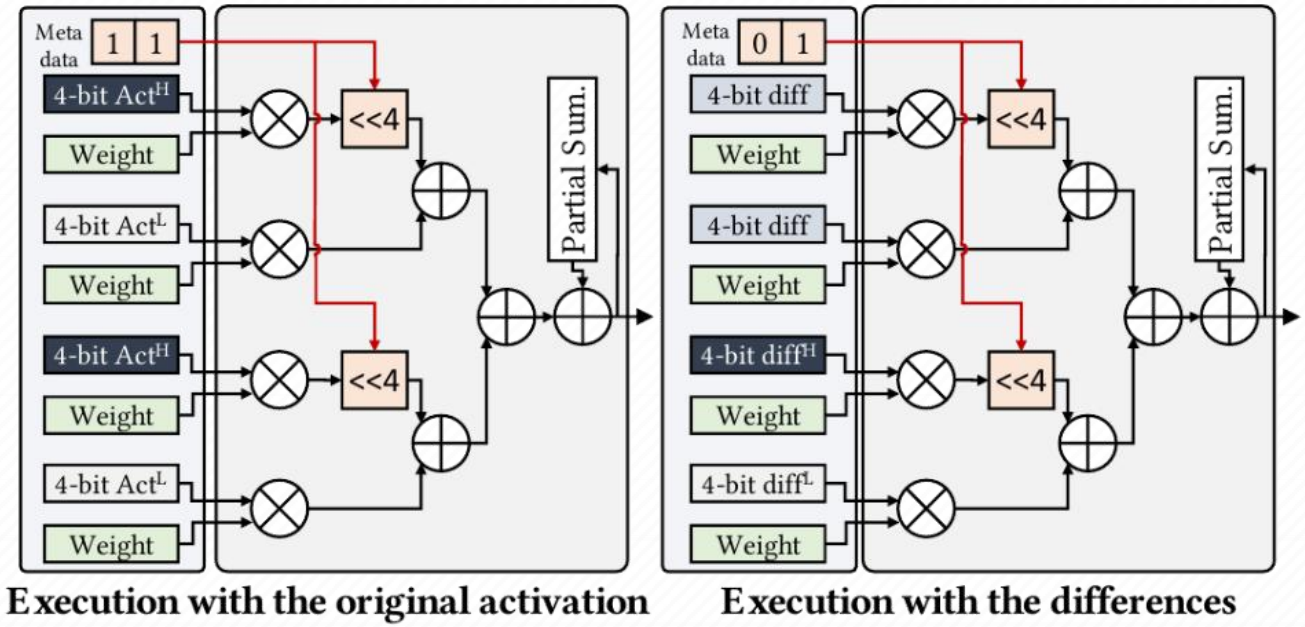
Hardware--Encoding Unit

- Calculating differences
- Classifying the bit-width requirement of data
- Reordering data for zero skipping and notation of full bit-width data.



Hardware--Compute Unit

- Support two types of bit-width, 8-bit full bit- width and 4-bit low bit-width data.
- Each PE consists of four multipliers that execute a multiplication between 4-bit data and weight, and a corresponding adder tree. To support 8-bit operations, shifters are applied in the first adder stage.



Results

TABLE II: Accuracy of Diffusion Models. FID is lower the better. IS and CS are higher the better.

Model	Metric	FP32	Ditto
DDPM	FID / IS	4.143 / 9.084	4.406 / 9.288
BED	FID / IS	2.962 / 2.227	5.897 / 2.338
CHUR	FID / IS	4.100 / 2.715	3.743 / 2.714
IMG	FID / IS	14.332 / 368.302	14.156 / 358.580
SDM	FID / IS / CS	20.547 / 37.345 / 0.310	18.834 / 38.135 / 0.309
DiT	FID / IS	18.659 / 482.372	17.178 / 475.694
Latte	IS	70.589	71.254

TABLE III: Hardware Configurations of Baseline and the Ditto Hardware

Hardware	# of PE	Bit-width of PE	Power (W)	SRAM (MB)	Area (mm^2)	Freq.
ITC [63]	27648	A8W8	36.9	192	64.48	1GHz
Diffy [58]	39398	A4W8	33.6			
Cambricon-D [43]	normal-38280 outlier-2552	A4W8 A8W8	33.3			
Ditto	39398	A4W8	33.6			

02

CMC: Video Transformer Acceleration via CODEC Assisted Matrix Condensing

Preliminaries

The l-th attention block receives the patch embedding vector $z^{(l-1)}$ as input and sequentially conducts temporal and spatial multi-head self-attention (MSA) modules, which follows Eqn. (1)-(2).

$$y_t^l = MSA_t(z_t^{(l-1)}) + z_t^{(l-1)} \quad (1)$$

$$y_o^l = MSA_s(y_s^l) + y_s^l \quad (2)$$

$$z^l = MLP(y_o^l) + y_o^l \quad (3)$$

where $z_t^{(l-1)}$ is formed by N input matrices, each containing T tokens extracted from the same spatial index of $z^{(l-1)}$. Similarly, y_s^l consists of T input matrices, each containing N tokens extracted from the same temporal index of y_t^l. As

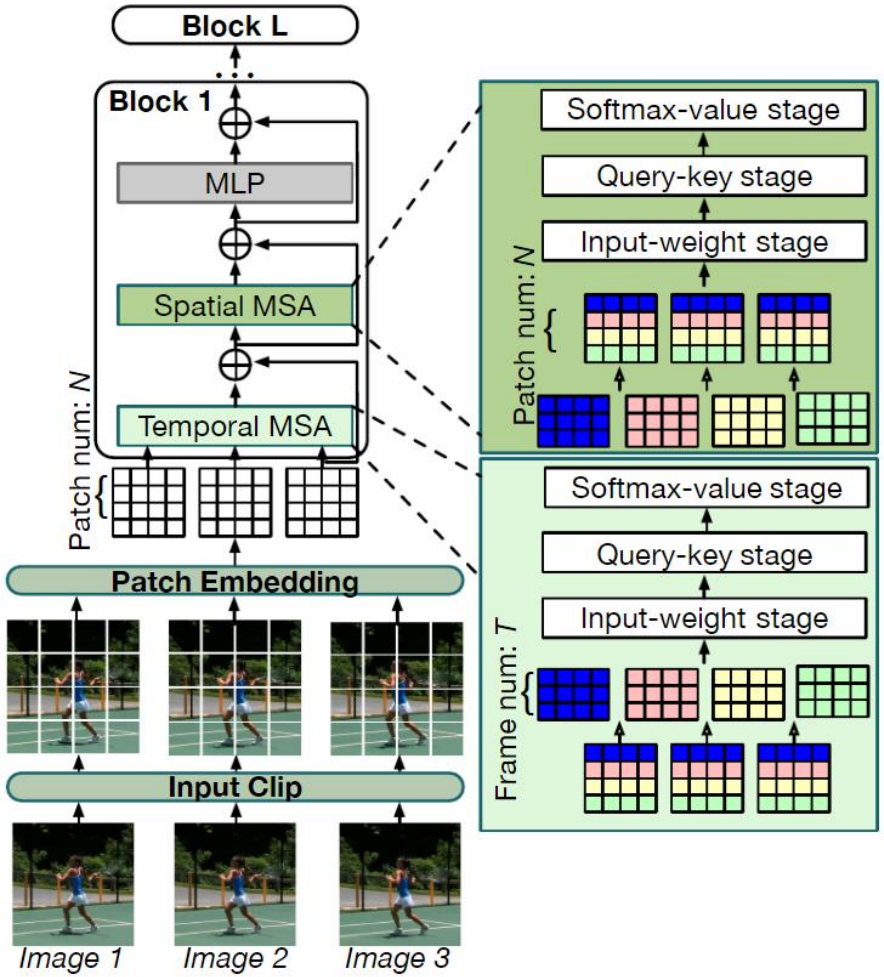
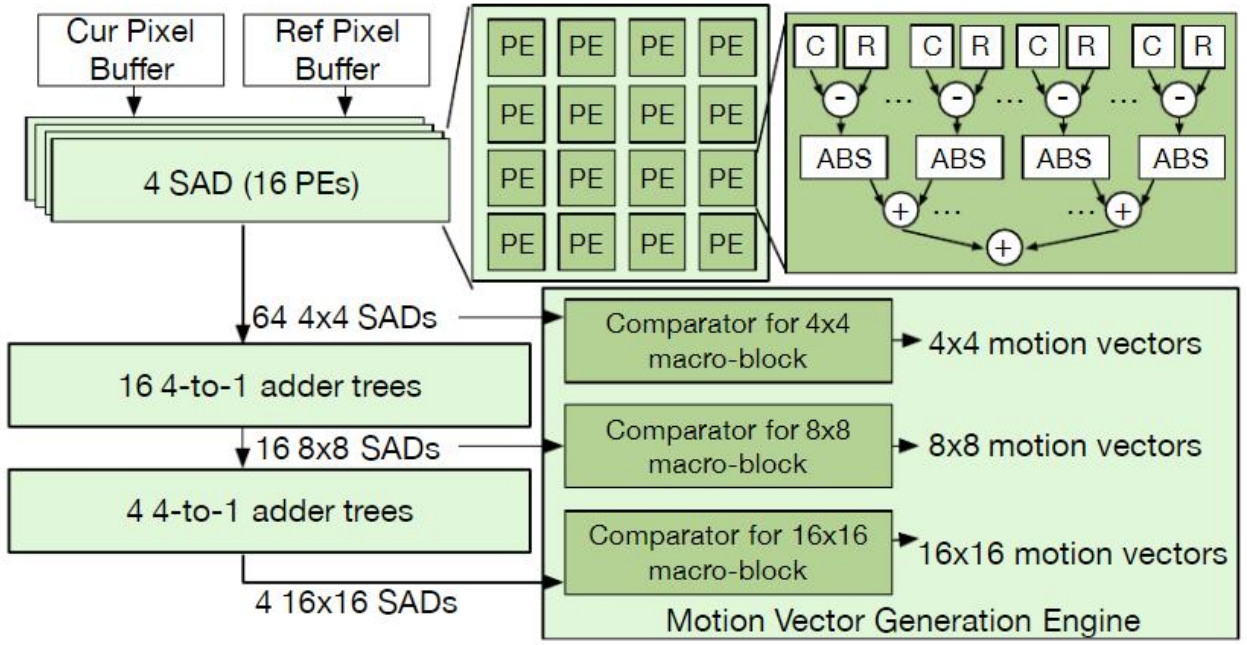


Figure 2. Overview of VidTs.

Preliminaries

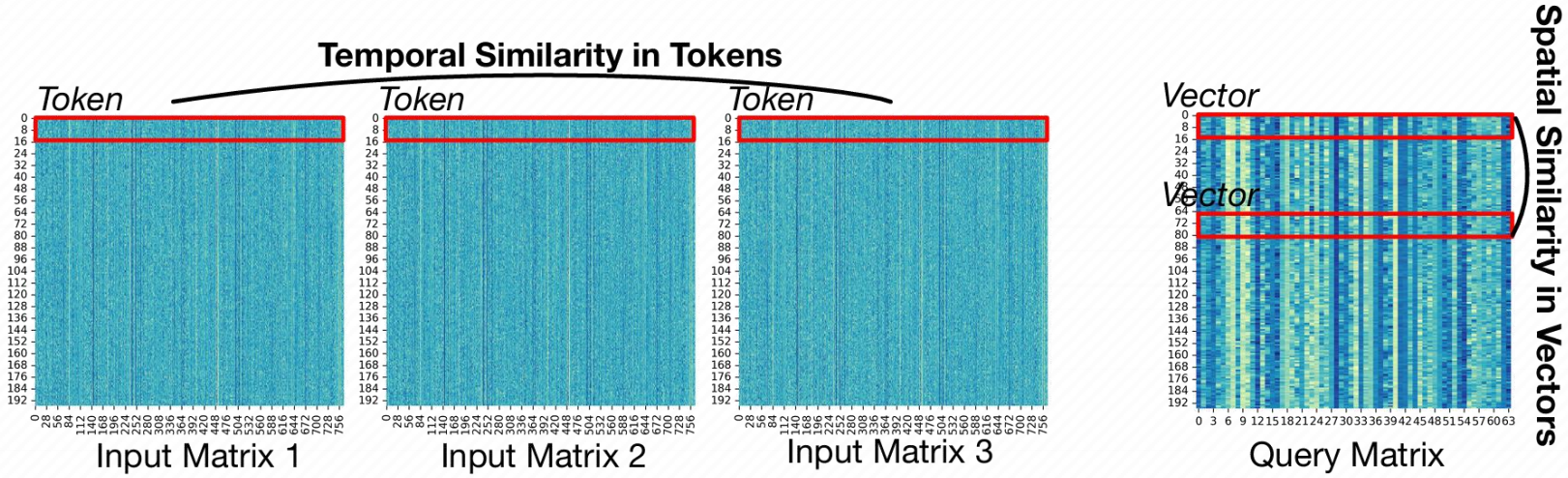
- Classify a series of raw frames into I and P frames
- For an I frame, each macro-block undergoes intra-frame prediction using 14 prediction modes.
- For a P frame, it utilizes the motion estimation (ME) algorithm to search for the most similar macro-block in previous frames.

$$SAD = \sum_i \sum_j |C(i,j) - R(i,j)|$$



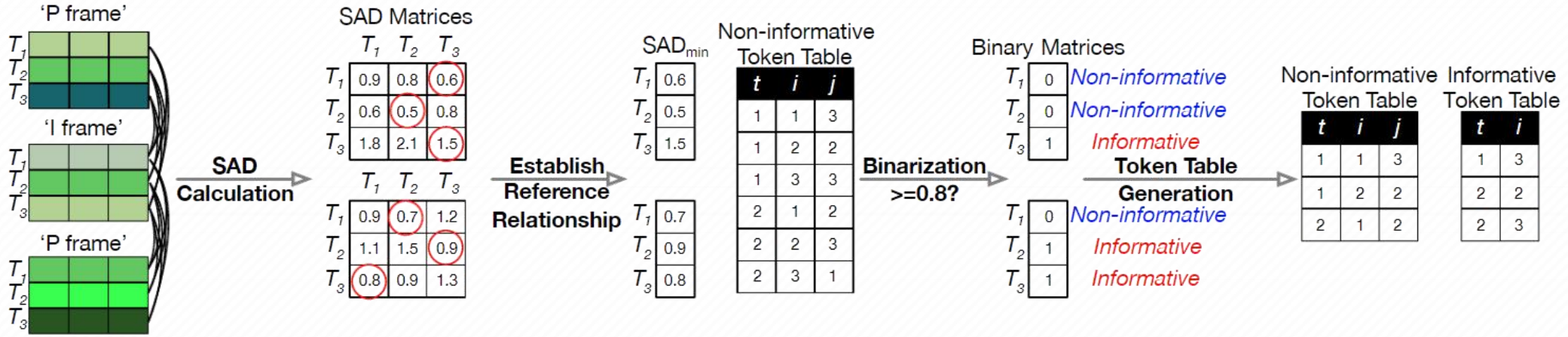
. The ME module in the video CODEC.

Motivation



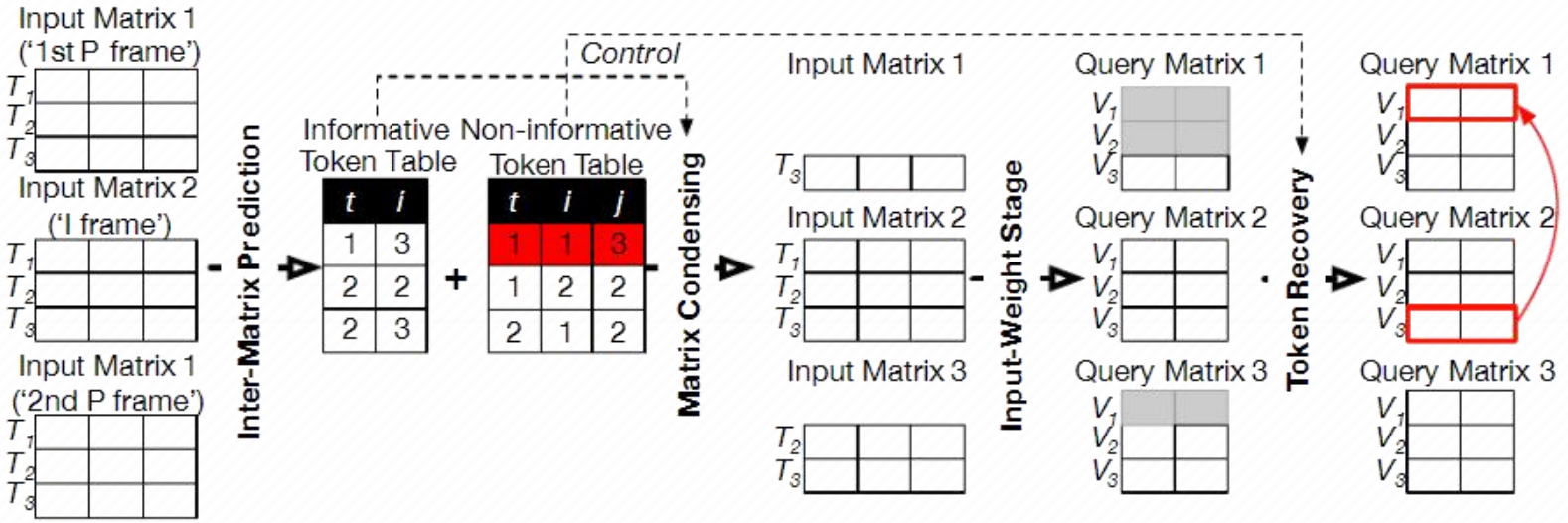
- The video data contains lots of repeated visual pixels, exhibiting temporal and spatial redundancies.

Method-Inter-Matrix Prediction



- In each segment, the middle input matrix is designated as the 'I frame'. All tokens within the 'I frame' are considered as informative tokens
- The remaining n - 1 matrices are treated as 'P frames', and the tokens in them are placed in a candidate buffer, awaiting prediction to determine their informativeness.
- Perform a calculation of SAD between tokens in 'P frames' and those in 'I frame' to measure their similarity. Tokens in 'P frames' that exhibit large SAD values are identified as informative tokens.

Method



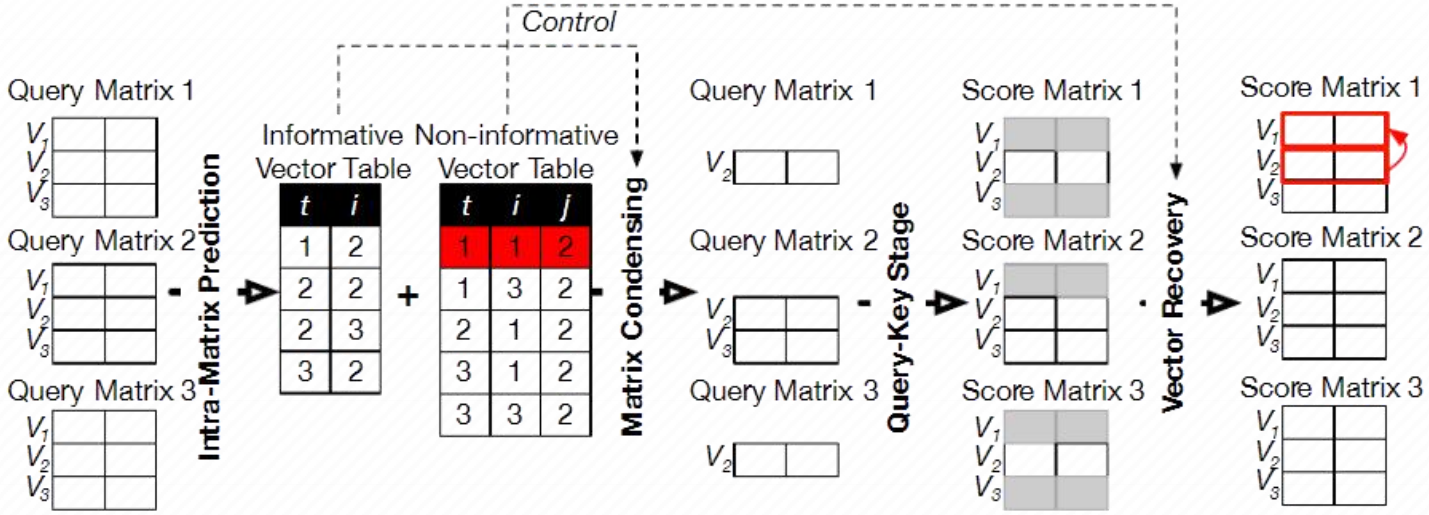
- first calculate the SAD between token Ti in a 'P frame' and token Tj in the 'I frame'.

$$SAD_{i,j}[t] = \sum_{v=1}^{V} \left| T_i^v[t] - T_j^v \right|$$

- Find the min SAD of Ti in a 'P frame' and set the corresponding token of 'I frame' as reference token.we represent this reference relationship as (t,i,j). These motion vectors collectively constitute a non-informative token table.

- apply a predefined threshold ThrT to binarize SADmin.

Method-Intra-Matrix Prediction

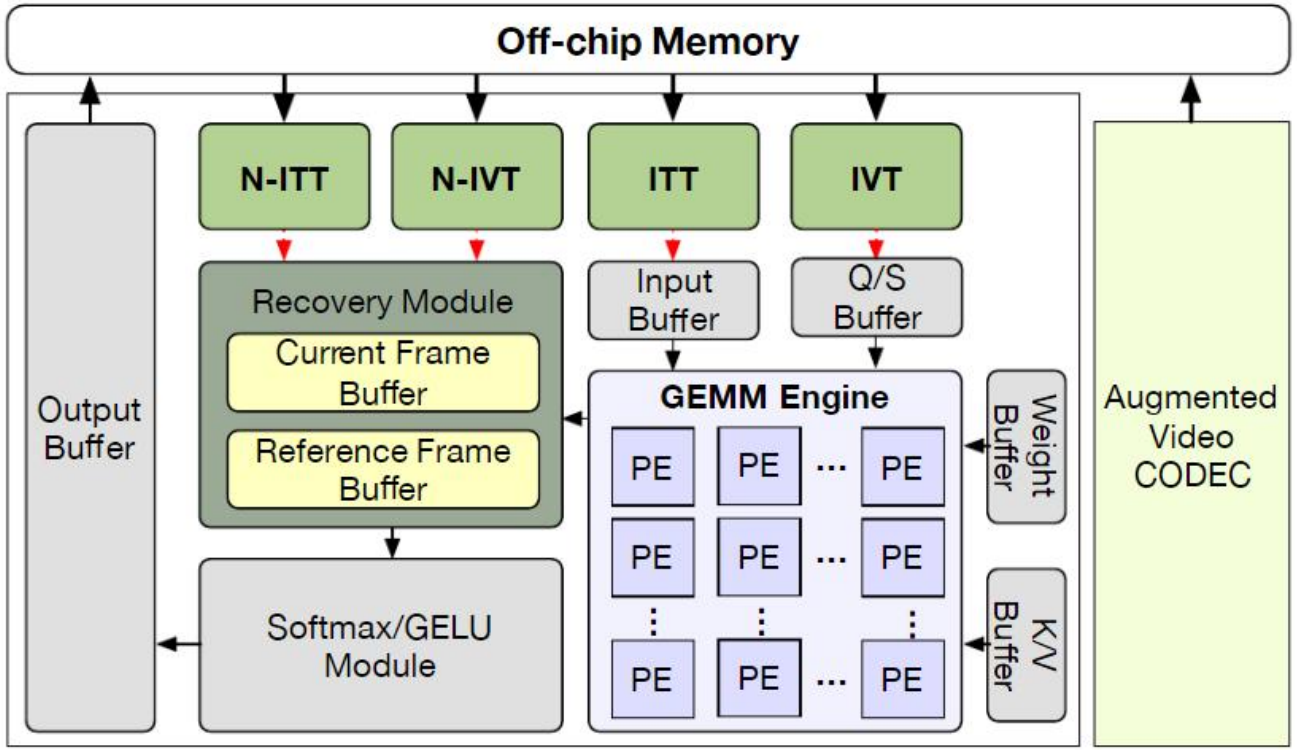


- Divide the matrix into segments and specify ‘I frame’ and ‘P frames’ within each segment.

- Calculate the SAD between vectors in the ‘I frame’ and ‘P frames’ to predict informative vectors on-the-fly. The informative vectors’ ID and the reference relationship between non-informative and informative vectors are respectively stored in the informative and non-informative vector tables.

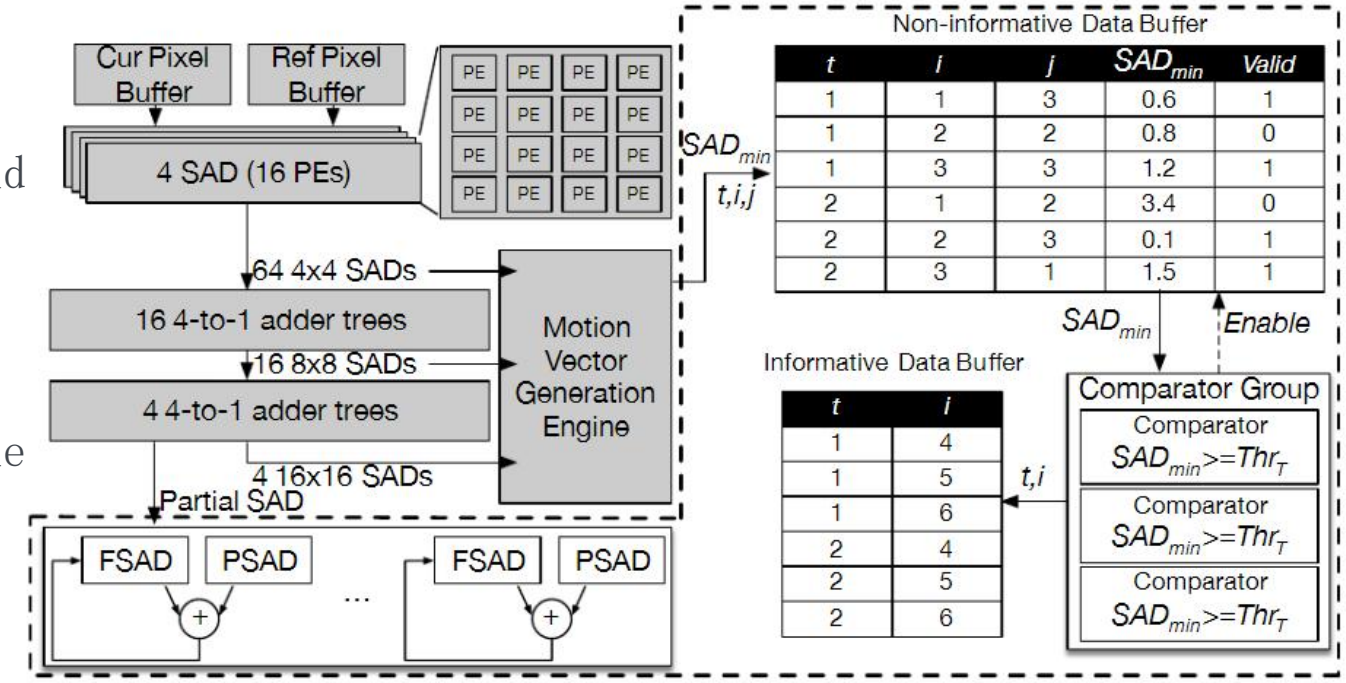
Full Architecture

- Informative token table (ITT);Informative vector table (IVT);Non-informative token table (N-ITT);Non-informative vector table (N-IVT)
- The augmented CODEC dynamically predicts informative tokens.
- Only the informative tokens are fetched into the input buffer.
- GEMM engine executes a relatively lightweight input-weight stage, which generates condensed query, key, and value matrices.
- The recovery engine then reconstructs the complete query, key, and value matrices and stores them back to DRAM.

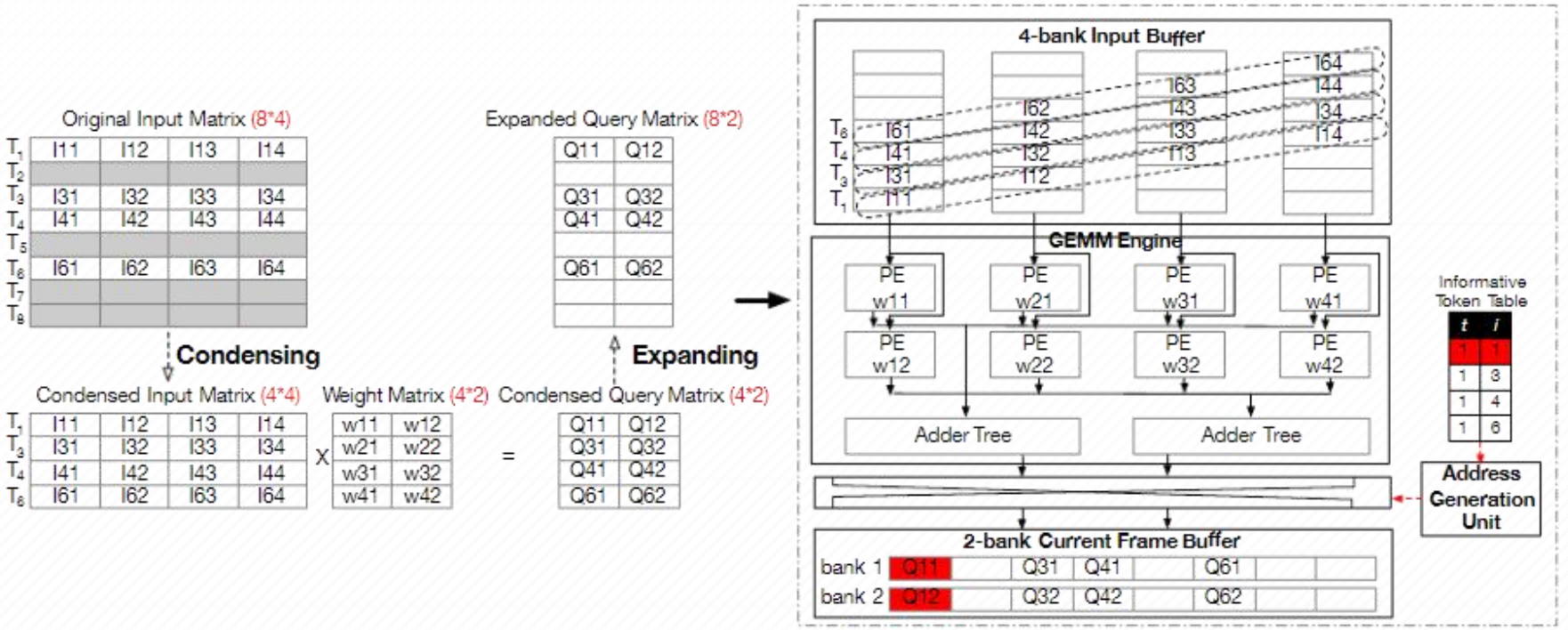


Augmented CODEC Design

- For tokens with a dimension n larger than the largest dimension m supported by the CODEC, invoke the CODEC ⌈n/m⌉ times. Sequentially compute the partial SAD of tokens and accumulate the results to obtain the final SAD.
- In the case where n < m, pad the token with zeros to align it with the macro-block's dimension.



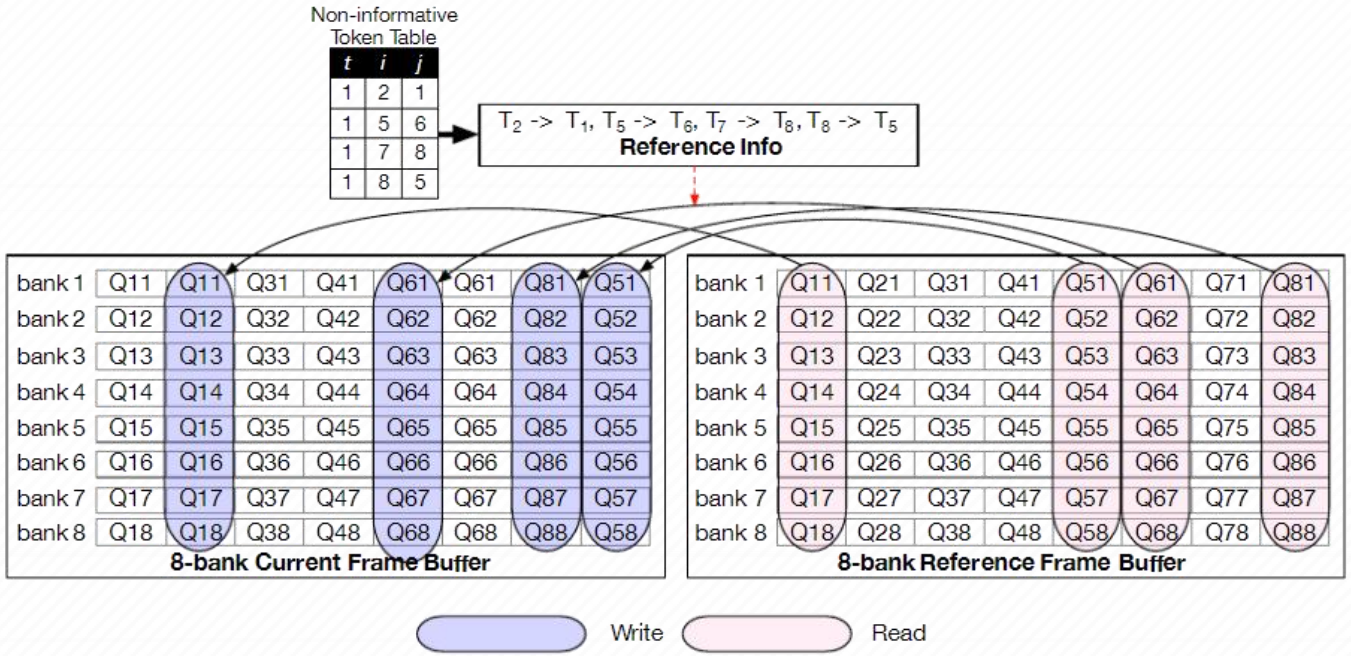
Fine-grained Buffer Management

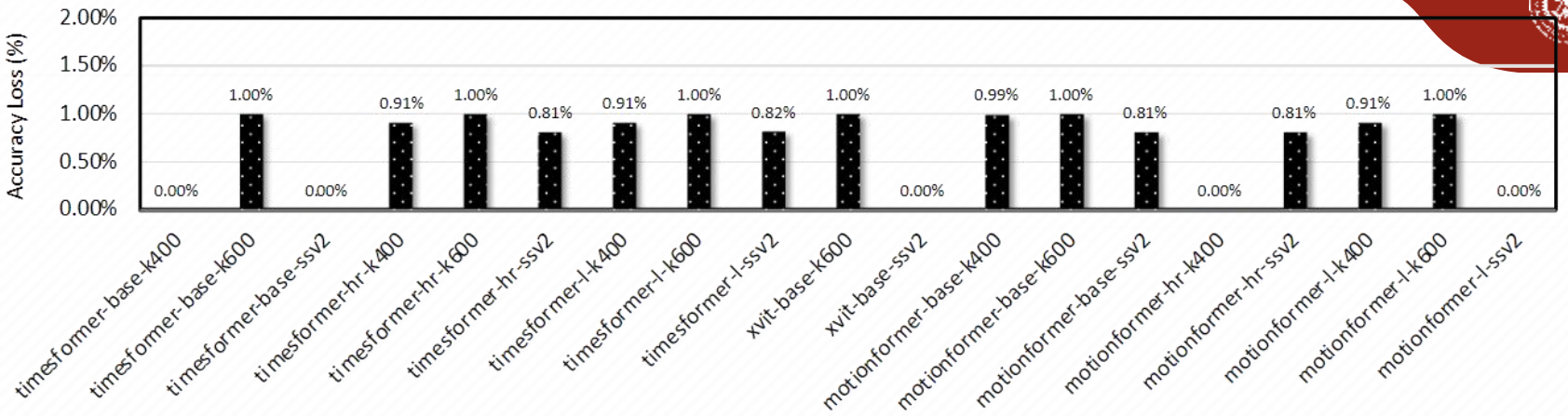


- The informative token table and the address generation unit manage the process by determining the bank ID and entry ID of each result.
- Rule: the j th adder tree that is responsible for accumulating the multiplication results of j th PE row should scatter the result to (bank j , entry i) given the streamed token ID i in the informative token table.

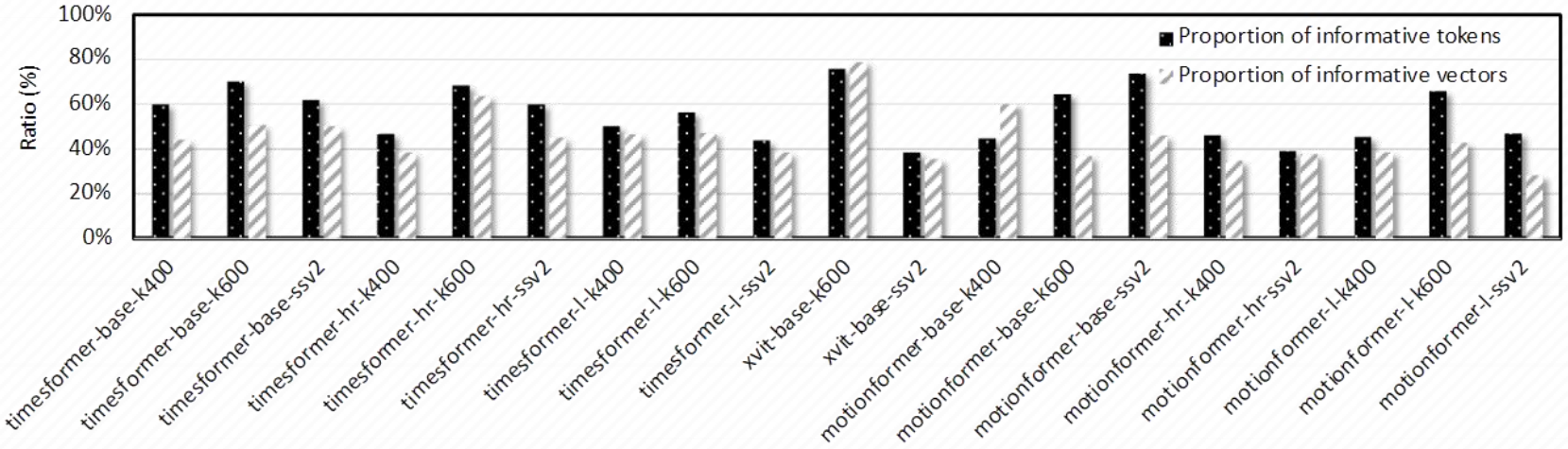
Recovery Module

- The recovery module reads a whole entry of multiple banks, corresponding to an informative token, from the reference frame buffer
- Write it back to a corresponding entry in the current frame buffer.
- To avoid time-consuming serial copying, Propose to pipeline the read and write operations.





(a)



(b)

Figure 11. Model accuracy (a), the proportion of informative tokens/vectors (b).

上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

THANKS FOR ALL

感谢您的观看与聆听！

SHANGHAI JIAO TONG UNIVERSITY