PARO: Hardware-Software Co-design with Pattern-aware Reorder-based Attention Quantization in Video Generation Models

Xinhao Yang*[1,2], Tianchen Zhao*[1,2], Hongyi Wang[1,2], Wenheng Ma[1,2], Shulin Zeng[1,2], Zhenhua Zhu[1],
Xuefei Ning[1], Huazhong Yang[1], Yu Wang[1†]
[1]Dept. of EE, BNRist, Tsinghua University [2]Infinigence-AI
[†]Corresponding author: yu-wang@tsinghua.edu.cn

***Abstract*—Transformer-based video generation models have demonstrated significant potential in content creation. However, the current state-of-the-art model employing "3D full attention" encounters substantial computation and storage challenges. For instance, the attention map size for CogVideoX-5B requires 56.50 GB, and generating a video of 49 frames takes approximately 1 minute on an NVIDIA A100 GPU under FP16. Although model quantization has proven effective in reducing both memory and computational costs, applying it to video generation models still faces challenges in preserving algorithm performance while ensuring efficient hardware processing. To address these issues, we introduce PARO, a video generation accelerator with pattern-aware reorder-based attention quantization. PARO investigates the diverse attention patterns of 3D full attention and proposes a novel reorder technique to unify these patterns into a unified "block diagonal" structure. Block-wise mixed precision quantization is further applied to achieve lossless compression under an average bitwidth of 4.80 bits. In terms of hardware, to overcome the limitation of existing mixed-precision computing units could not fully utilize the attention map bitwidth to accelerate QK multiplication, PARO designs an output-bitwidth aware mixed-precision processing element (PE) array through hardware-software co-design. This approach ensures that the mixed-precision characteristics are fully utilized to enhance hardware efficiency in the bottleneck attention computation. Experiments demonstrate that PARO delivers up to 2.71× improvement in end-to-end performance compared to an NVIDIA A100 GPU and achieves up to 6.38∼7.05× speedup over state-of-the-art ASIC-based accelerators on the CogVideoX-2B and 5B models.**

***Index Terms*—Video Generation Model, Mixed-precision Quantization, Hardware Accelerator.**

I. Introduction

Diffusion Transformers (DiTs) [1] and video generation models [2] have garnered significant research interest after the impressive generation quality of OpenAI's SORA [3] in 2024. Previous researches, like OpenSORA [2], utilize "spatial-temporal" attention, which performs attention separately along the spatial and temporal dimensions. More recent models such as CogVideoX [4] adopt "3D full attention", which processes all the spatial tokens for all frames together. Such an attention scheme further enhances algorithm performance, thus achieving state-of-the-art video generation quality. However, such an attention scheme leads to an order of magnitude increase in the token length for attention computation, resulting in excessive storage and computation costs. For example, the token length is 17.8k for the CogVideoX model, and the attention map takes up 56.50 GB for each transformer block. The attention computation accounts for 67.93% of the overall latency on an NVIDIA A100 GPU, which becomes the major bottleneck.

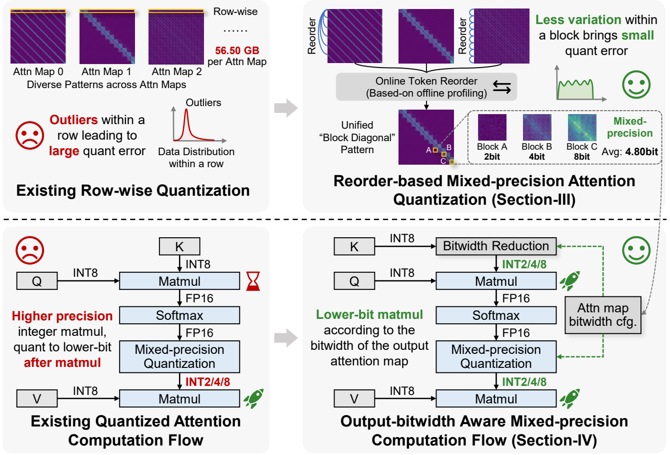
Attn Map 0
Attn Map 1
Attn Map 2
Row-wise
......
56.50 GB
per Attn Map
Diverse Patterns across Attn Maps
Outliers within a row leading to large quant error
Outliers
Data Distribution within a row
Existing Row-wise Quantization
Reorder
Reorder
Online Token Reorder
(Based-on offline profiling)
Less variation within a block brings small quant error
Unified "Block Diagonal" Pattern
A B C D
Block A 2bit
Block B 4bit
Block C 8bit
Mixed-precision
Avg: 4.80bit
Reorder-based Mixed-precision Attention Quantization (Section-III)
K
Q
V
INT8
INT8
INT8
Matmul
FP16
Softmax
FP16
Mixed-precision Quantization
INT2/4/8
Matmul
Higher precision integer matmul, quant to lower-bit after matmul
Existing Quantized Attention Computation Flow
K
Q
V
INT8
INT8
INT8
Bitwidth Reduction
INT2/4/8
Matmul
FP16
Softmax
FP16
Mixed-precision Quantization
INT2/4/8
Matmul
Attn map bitwidth cfg.
Lower-bit matmul according to the bitwidth of the output attention map
Output-bitwidth Aware Mixed-precision Computation Flow (Section-IV)

Fig. 1. Overview of the challenges and solutions in PARO.

To mitigate computation and memory consumption, model quantization is a highly effective technique that replaces high-precision floating-point weights and activations with low-bit integers. Low-bit integer multiplications are more hardware-efficient compared to high-precision floating-point operations. Although many existing researches have explored the quantization [5]–[10] of Vision Transformers (ViT) [11], directly applying existing methods to the models with 3D full attention causes notable degradation under INT8 and failure under INT4. Moreover, existing hardware accelerators can only benefit from the low-bit input tensors, failing to efficiently handle the scenario of computing the low-bit (e.g., 2bit) attention map from Q, K embeddings in higher-bits (e.g., 8bit), which is the bottleneck of 3D full attention video generation models.

To address these challenges, we delve into the data distribution characteristics of the attention maps. As illustrated in the upper left part of Fig. 1, we observe diverse "diagonal" patterns across different attention heads and transformer blocks. These patterns arise from the local information aggregation properties inherent in vision feature extraction. Traditional quantization methods assign the same set of quantization parameters to each row of the attention maps. However, the larger elements on the diagonal act as "outliers", leading to excessively large quantization scaling factors. In these cases, the majority of values are forcedly set to near-zero values,

*Both authors contributed equally to this work.

thus losing discriminative abilities, and resulting in large quantization errors. To mitigate this, we propose a reorder-based mixed-precision quantization method. By reordering the QK embeddings along the token dimension, we transform the diverse patterns into a unified "block diagonal" pattern and apply block-wise quantization. This reorder effectively clusters similar values into local blocks, reducing data variation within these blocks and thereby minimizing quantization errors. Moreover, different blocks show diverse attention values, contributing differently to the final attention outputs. Therefore, to compress for lower bits, we introduce mixed-precision quantization, which assigns higher bitwidths to blocks with larger attention values and quantization difficulty, thereby optimizing the overall quantization accuracy and efficiency.

In terms of hardware efficiency, existing mixed-precision computing units perform multiplication and accumulation (MAC) operations based on the bitwidths of the input tensors. These units are unable to leverage the mixed-precision of the output to reduce computational workload. Taking attention computation as a typical example, the attention map, as discussed earlier, can be quantized to mixed precision (e.g., 2/4 bits), while the QKV embeddings remain 8bit. For the $AttnV$ computation, the mixed-precision characteristics of the attention map can be effectively utilized by existing mixed-precision computing units. However, for $QK^\top$, despite the lower precision of the output attention map, existing computing units are constrained by the input bitwidths and must perform the matrix multiplication in 8bits. This limitation prevents them from exploiting the lower bitwidths of the output to further improve computational efficiency. To address this, with a pre-determined lower bitwidth configuration of the output tensor, the matrix multiplication can be performed in reduced bitwidths, aligned with the output precision. Unlike current methods that optimize only $AttnV$ [10], [12], this approach enables the acceleration of both $QK^\top$ and $AttnV$ computations, with negligible accuracy loss. By further exploiting the acceleration opportunities offered by mixed-precision attention maps, this method boosts the efficiency of attention processing, which is the performance bottleneck.

The contributions of PARO are as follows:

- We identify the key limitation of existing quantization methods by investigating the unique properties of "3D full attention" in video generation models, and design a reorder-based mixed-precision quantization method tailored for them.
- To fully harness the performance gain of mixed-precision quantization schemes, we design output-bitwidth aware mixed-precision processing elements (PEs). It fully exploits the capability of PEs to adapt to mixed-precision cases for both QK^T and $AttnV$ in attention computation.
- PARO achieves lossless quantization with an averaged 4.80bit and outperforms state-of-the-art ViT accelerator ViTCoD by 6.38∼7.05×. Evaluated under the same hardware resources, PARO achieves a 1.68∼2.71× speedup to an NVIDIA A100 GPU.

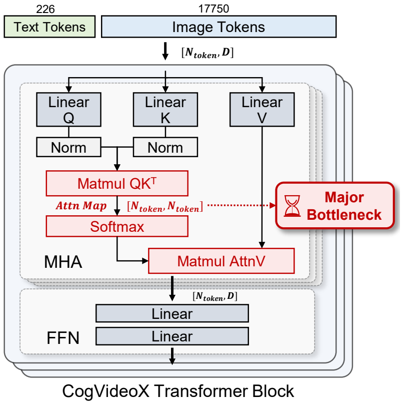

Fig. 2. Illustration of CogVideoX text-to-video generation model, in which the major bottleneck is the attention map related computations.

II. Background and Related Work

A. Video Generate DiTs

As discussed in Sec. I, our primary focus is to optimize the state-of-the-art video generation model, which utilizes a 3D full-attention mechanism, specifically the CogVideoX model. The model structure of CogVideoX is illustrated in Fig. 2, it comprises 42 transformer blocks. Each transformer block consists of two key components: multi-head self-attention (MHA) and feed-forward network (FFN). The computation of the MHA could be formulated as:

$$Q = XW_Q, K = XW_K, V = XW_V, O = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V$$

Given an input token sequence with hidden dimension d, the attention projects it into query Q, key K, and value V matrices through linear layer with weights W_Q, W_K, W_V. Then, softmax is applied to the dot product of Q and K, divided by $\sqrt{d}$ to generate the attention map. Finally, the attention map is multiplied with V to generate the attention output. Subsequently, the FFN processes the attention output O with several full-connected linear layers. In CogVideoX, N_{token} is 17.8k, which is significantly larger than the hidden dimension (d), ranging from 1k to 4k. The computation of the attention map, which has a shape of $[N_{token}, N_{token}]$, becomes the major bottleneck (highlighted in red in Fig. 2).

B. DiT Quantization

Model quantization [13], [14] has been demonstrated as an effective method for model compression. By converting high bitwidth floating-point (FP) data into lower bitwidth integers, it significantly reduces both computational and memory costs. In this approach, weights and activations are quantized within each group G (e.g., tensor-wise or channel-wise). The quantization process approximates the FP value x using an integer representation x_{int} and quantization parameters (scaling factor s, zero point z):

$$x \approx \hat{x} = s(x_{\text{int}} - z)$$

For a group of size g, represented as the vector $x \in \mathbb{R}^g$, all elements share the same quantization parameters (s and z). The quantization operator Q with b-bit is described as:

$$x_{\text{int}} = Q(x; s, z, b) = \text{clamp}(\left\lfloor \frac{x}{s} \right\rceil + z, 0, 2^b - 1)$$

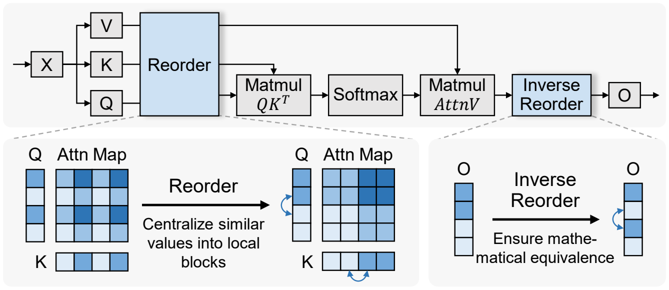

Fig. 3. Detailed description of the reorder process. Q, K, V are reordered along the token dimension to form a unified "block diagonal" pattern. The attention output O is inversely reordered to ensure mathematical equivalence.

In the quantization schemes used in recent work, weights are quantized offline, while activations are dynamically quantized online using a scaling factor $s = \frac{\max(x)-\min(x)}{2^b-1}$, where b is the bit width. The granularity for the quantization of weight and activation in linear layers is "per dimension" and "per token", respectively. For attention map quantization, the granularity is "per-row" for attention values and "per-dimension" for V.

C. Related Work

DiT Quantization. Existing methods [15] for DiT quantization focus on compressing the linear layers in video generation models with spatial-temporal attention. SageAttention and its follow-up work [16]–[18] further advance this by employing 8-bit quantization for QK in attention map computation to accelerate ”3D full attention.” In our method, we extend quantization to the linear layers, QKV, and attention maps, aiming to minimize expensive floating-point computations.

Hardware Accelerators. Extensive researches [19]–[24], have focused on designing customized accelerators for attention mechanisms. Sanger [25] introduces a locally structured sparse approach to reduce computational overhead. However, this method faces significant challenges when applied to video generation models. ViTCoD [26] addresses this issue with a threshold-based dynamic sparsification technique specifically designed for the attention mechanism in video generation models. It partitions the attention matrix into sparse and dense segments, processing them separately to improve efficiency. Other works [20], [24], [27] have explored strategies such as token pruning and joint video encoding to accelerate entire models. Despite these advancements, two major limitations persist for 3D attention mechanisms. First, existing methods fail to leverage the inherent similarities in 3D attention patterns, limiting opportunities for further optimization and compression. Second, their coarse-grained compression techniques struggle to achieve a balance between maintaining high accuracy and delivering significant acceleration, particularly without requiring additional model retraining.

III. Reorder-based Attention Quantization

A. Reorder-based Block-wise Quantization

In the popular 3D full attention models (e.g., CogVideoX), the token length (N_{token}) is significantly larger than the hidden dimension (d), leading to attention computation dominating the overall computational cost due to its quadratic complexity with respect to token length. As such, quantizing the attention map is critical for resource efficiency. However, attention quantization presents unique challenges. SageAttention, for instance, only quantizes the Q and K tensors, which accelerates only half of the attention computation.

In this paper, we extend this approach by quantizing Q, K, V, and the attention map. We begin by adopting a naive quantization scheme, adopting dynamic min-max quantization, using row-wise grouping for the attention map and dimension-wise grouping for V. However, we observe severe quality degradation under INT8 when solely quantizing the attention map. Additionally, the INT4 results in unreadable noise outputs. To investigate the causes of these issues, we visualize the attention map. As shown in the upper left part Fig. 1, the data within each quantization group (a row in the attention map) exhibits significant variation. A small subset of values, or "outliers", are significantly larger than the rest. The scaling factor (s) is determined based on these outliers, resulting in an excessively large scaling factor for most elements and introducing significant quantization errors. Based on these findings, we aim to reduce quantization errors by minimizing data variation within each quantization group.

We further visualize and analyze the structure of attention maps and observe distinct patterns across different blocks and heads, as illustrated in the upper left part of Fig. 1. An intuitive approach is to leverage these patterns to design specialized quantization groupings that reduce data variation within groups. However, the diversity of these patterns introduces challenges in designing distinct groupings for each case. To address this, we propose transforming these diverse patterns into a unified, hardware-friendly format to enhance hardware efficiency. Through detailed analysis, we find that all patterns essentially represent local aggregation across different dimensions (e.g., the same token across frames or neighboring spatial tokens). By reordering tokens, we can reorganize these patterns into local blocks, as shown in the upper right part of Fig. 1(b). Using block-wise grouping for quantization minimizes variation within each block, thereby reducing quantization error and improving overall performance.

Given an input token of size $N_{\text{frame}} \times N_{\text{width}} \times N_{\text{height}}$, we achieve local block-wise patterns by permuting these dimensions for the QK embeddings through token-level reorder. There are a total of 6 possible reorder plans for each attention head. Notably, the observed patterns remain consistent across different timesteps and input noise or prompts. We select the reorder plan that minimizes quantization error for each head and block offline. The reorder itself is performed online. Due to the compute-bound nature of the model, the overhead introduced by reorder is negligible.

B. Importance-guided Mixed Precision

As shown in the upper right part of Fig. 1, after reordering, similar values are grouped into localized blocks, however, different blocks still exhibit varying value distributions and contribute differently to the final results. These blocks have diverse "quantization sensitivities". Consequently, applying the

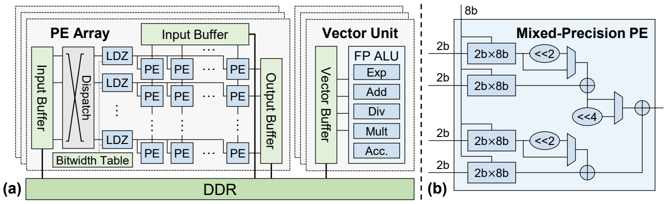

Fig. 4. (a) Overall architecture of PARO. (b) Each PE supports three multiplication mode: 2bit×8bit, 4bit×8bit, 8bit×8bit.

same bitwidth to all blocks results in a suboptimal trade-off between algorithm quality and hardware efficiency. A natural solution is to adopt mixed-precision quantization, allocating higher bitwidths to blocks with higher sensitivity.

Accurately estimating the "quantization sensitivity" is crucial for effective mixed-precision bitwidth allocation. We identify two key characteristics of data distribution that influence quantization sensitivity: (1) **"Block Importance"**: The average absolute values of each block vary, reflecting their relative significance for attention values. (2) **"Quantization Difficulty"**: Within each block, the degree of data variation differs, representing the block's "quantization difficulty". Both highly important blocks and those with larger quantization difficulty should be assigned higher bitwidths. Considering these factors, we propose the following sensitivity metric:

$$\mathcal{S} = (\sum_{i}^{G} x_i)^{\alpha} * (||x_i - x_q||)^{(1-\alpha)}, x \in R^G$$

For each block containing G values, we employ the average attention value as the "importance" and the quantization error to represent the "quantization difficulty". The hyper-parameter α serves to balance the relative emphasis between these two factors. Then, we formulate mixed precision bitwidth allocation as an integer programming problem as follows:

$$\begin{aligned} &\underset{c_{i,b}}{\text{argmin}} \quad \sum_{i=1}^{N} \sum_{b=0,2,4,8} c_{i,b} \cdot \mathcal{S}_{i,b} \\ &\text{s.t.} \quad \sum_{b=0,2,4,8} c_{i,b} = 1, \quad \sum_{i=1}^{N} \sum_{b=0,2,4,8} c_{i,b} \cdot b \leq \mathcal{B} \cdot N, \\ &\quad c_{i,b} \in \{0,1\}, \quad \forall i \in \{1, \cdots, N\}, \forall b \in \{0,2,4,8\} \end{aligned} \tag{1}$$

where N is the number of blocks in the model; $c_{i,b} = 1$ indicates that the i-th block will be quantized to b-bit, and $\mathcal{S}_{i,b}$ is the corresponding sensitivity score, $\mathcal{B}$ indicates the average bitwidth budget.

IV. PARO Architecture

A. Architecture Overview

The overall architecture of PARO is shown in Fig. 4(a), comprising multiple PE arrays, vector units, DDR, and a controller (not shown in the figure). Among these components, the fixed-point PE arrays serve as the primary fixed-point computation unit of the architecture, executing all matrix multiplications since PARO applies quantization to all linear and attention layers. The vector unit features a floating-point arithmetic and logic unit (ALU) and handles floating-point computations outside matrix multiplications (e.g., softmax).

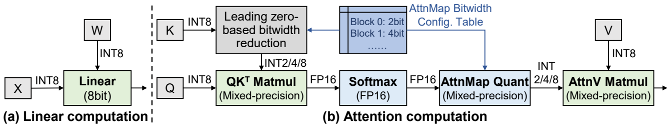

Fig. 5. (a) For linear computations, matrix multiplication operates in 8bit precision. (b) For attention computations, the bitwidth is dynamically reduced based on the attention map's bitwidth configuration.

As the quantization scales for linear and attention layers are in FP16 format, fixed-point accumulation results computed by the fixed-point PE arrays are forwarded to the vector unit. The vector unit converts these results to FP16 format and performs floating-point accumulation, yielding the final FP16 output for matrix multiplication.

B. Mixed-precision PE Array

The structure of each PE in the mixed-precision PE array is shown in Fig. 4(b). Each PE consists of four 2bit×8bit fixed-point multipliers. By adjusting the control signals of the multiplexers, each PE can execute one 8bit×8bit, two 4bit×8bit, or four 2bit×8bit multiplications per cycle.

For linear layers with W8A8 quantization, the PE array operates in 8bit×8bit mode, as shown in Fig. 5(a). In contrast, for attention layers, the bitwidth varies across different blocks in the attention map, while the QKV matrices remain 8bit. Since $QK^\top$ and $AttnV$ each account for half of the computations in attention, the PE array must efficiently handle two scenarios:

- $AttnV$: One input matrix (attention map) is mixed-precision while the other (V) is 8bit.
- $QK^\top$: Both input matrices (Q, K) are 8bits, but the output matrix after softmax is quantized to mixed-precision.

For $AttnV$, computations can be directly mapped to the PE array by dynamically configuring the PE mode based on the bitwidth table of each block in the attention map. For $QK^\top$, performing matrix multiplication at the bitwidth of the inputs is often computationally expensive because the output may be quantized to extremely low bitwidths (e.g., 2bit) after softmax. In extreme cases, such as when the attention map block has a bitwidth of 0, the computation for that block can be skipped entirely. To reduce computational costs, the matrix multiplication can be performed at a lower precision that approximates the same result after softmax quantization.

Inspired by this, PARO adopts an output-bitwidth-aware mixed-precision computation flow to leverage the mixed-precision characteristics of the output attention map. As illustrated in Fig. 5(b), this approach dynamically reduces the bitwidth of K to match the bitwidth of the corresponding output attention map block with little error. Experiments show that this optimization produced no perceptible differences in the quality of the generated video. By performing matrix multiplication at lower precision, the mixed-precision PE Array effectively accelerates $QK^\top$ computation.

Bitwidth reduction of K is implemented using a leading zero (LDZ) unit, integrated beside each PE row, as shown in Fig. 4(a). The LDZ unit identifies the most significant valid bit (MSVB) of the data and outputs the MSVB along with the following $K-1$ bits. The MSVB is the first 1 for positive

TABLE I

Algorithm performance of text-to-video generation on CogVideoX prompt set. The description of metrics is provided in Sec. V-A. We compare PARO quantization method with baseline methods, and ablates each technique.

Method	Block-wise	Reorder	Mixed Precision	Bitwidth	FVD-FP16 (↓)	CLIPSIM (↑)	CLIP-Temp (↑)	VQA (↑)	Flicker. (↑)
FP16	-	-	-	16	0.0	0.201	0.997	52.86	97.1
SageAttention [16]	-	-	-	8 (QK-only)	0.08	0.200	0.997	51.25	97.1
Sanger [25]	-	-	-	-	0.22	0.195	0.991	50.84	97.0
Naive INT8	-	-	-	8	0.44	0.201	0.998	49.80	97.0
Block-wise INT8	✓	-	-	8	0.21	0.203	0.997	52.42	97.3
PARO INT8	✓	✓	-	8	0.19	0.203	0.997	50.92	97.2
Naive INT4	-	-	-	4	1.40	0.187	0.997	16.79	96.4
Block-wise INT4	✓	-	-	4	0.40	0.201	0.998	46.53	96.9
PARO INT4	✓	✓	-	4	0.28	0.202	0.998	50.12	96.9
PARO MP	✓	✓	✓	4.80	0.15	0.205	0.998	52.61	96.9

values and the first 0 for negative values. For example, if the LDZ unit is configured as 2bit, the 8bit value `8b00011010` is compressed to `2b11`. After multiplication, the result is restored by left-shifting based on the bit index of MSVB.

The varying bitwidths (0/2/4/8) across different blocks result in differing throughputs for processing these blocks. Therefore, a dispatcher is integrated into each PE array to balance the workloads across blocks. Notably, PARO supports 0bit quantization blocks. The dispatcher bypasses computation for 0bit blocks and maps the next block to the appropriate PE row.

V. Evaluations

A. Evaluation Setup

Software Implementation. We apply the PARO quantization method, referred to as “PARO MP,” to the CogVideoX-5B model. In PARO MP, the weights and activations of all linear layers are quantized to INT8. For attention computation, $QKVO$ are quantized to INT8, while the attention map after softmax is quantized using mixed precision (0, 2, 4, 8 bits). Notably, ”0 bit” signifies skipping the computation for the corresponding block. Following the official implementation, we generate the 640×480, 49 frames videos, using CogVideoX example prompt set, DDIM [28] 50 steps. We evaluate the generation quality from various aspects following prior literature [15]. The “FVD-FP16” [29] estimate the fidelity of generated videos through measuring the feature space difference between quantized and FP16 outputs. The “CLIPSIM” [30] and “CLIP-Temp” [31] measures the text-video alignment and consistency of clip features across frames. The “VQA” [32] assesses the video quality from the aesthetic and technical perspective. The “Flicker.” measures the temporal flickering. higher value denotes less flickering. We present both the statistical and qualitative results in Tab. I and Fig. 7.

Platforms for Comparison. We evaluate our architecture against two state-of-the-art ASIC-based accelerators, Sanger [25] and ViTCoD [26]. The attention map is pruned following the methods described in their papers, ensuring that the generation quality and accuracy remain consistent with PARO. Additionally, we measure the performance of the NVIDIA A100 using CUDA Events for execution time and `nvidia-smi` for power consumption.

TABLE II

Area and power breakdown of PARO.

Component	Config	Area (mm^2)	Power (W)
PE Array	32×32×32 PEs	2.52 (30.8%)	3.60 (32.2%)
	Leading Zero Unit	0.65 (8.0%)	0.78 (7.0%)
	Others	0.39 (4.8%)	0.54 (4.8%)
Vector Unit	Exp/Div/Add/Mult/Acc.	2.79 (34.1%)	4.55 (40.6%)
Buffer	1.5 MB SRAM	1.82 (22.3%)	1.73 (15.4%)
Total	TSMC 12nm	8.17 (100%)	11.20 (100%)

Hardware Implementation. PARO is implemented using RTL for its hardware components and synthesized under TSMC 12nm process at 1 GHz using Synopsys Design Compiler to evaluate area and power. The hardware components are detailed in Tab. II. The DDR bandwidth of PARO is 51.2 GB/s. We use CACTI 7 [33] to assess the on-chip buffer in PARO. To ensure a fair comparison, we develop a cycle-accurate simulator to model the behavior and performance of both PARO and the baseline accelerators under the same hardware resource constraints. Furthermore, we align PARO’s hardware resources (e.g., peak computing performance, memory bandwidth, frequency, on-chip buffer size, etc.) with those of the NVIDIA A100 GPU to evaluate our performance gains over modern GPUs, denoted as “PARO-align-A100”.

B. Evaluation Results

Algorithm Performance. We evaluate the algorithm performance (generated video quality) against baseline methods, as detailed in Tab. I and Fig. 7. From both statistical and qualitative perspectives. “PARO-MP 4.80bit” demonstrates comparable or superior algorithm performance to baseline model compression techniques like “SageAttention” and “N:M sparse”. It notably outperforms “PARO INT4” in all aspects and achieves similar algorithm performance to “PARO INT8”. Compared to the FP16 baseline, “PARO-MP 4.80bit” incurs only a 0.25 VQA decrease and a 0.2 flickering score loss, which is hardly noticeable, as presented in Fig. 7.

Ablation Studies for Algorithm. We perform ablation studies for each component of the PARO quantization method, as shown in Tab. I. The “Naive” represents the naive round-to-nearest quantization scheme, and both “Naive INT8” and “Naive INT4” exhibit significant algorithm performance degra-

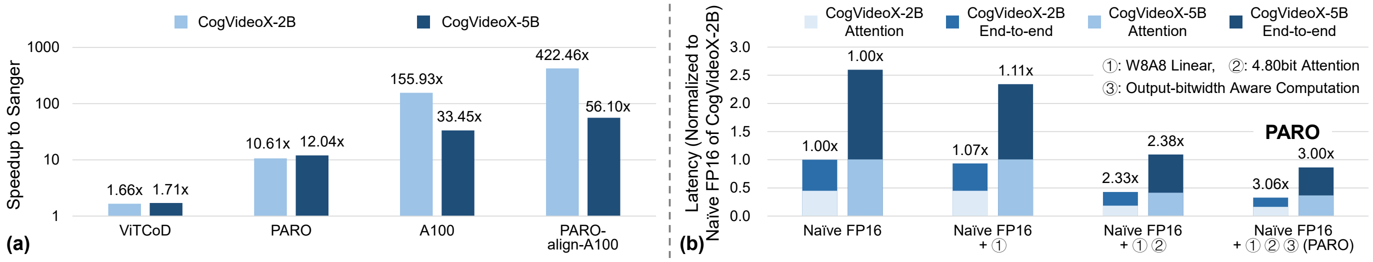

Fig. 6. (a) End-to-end speedup on CogVideoX-2B/5B, normalized to Sanger [25]. (b) The ablation study on different optimizations in PARO.

dation across multiple metrics. Incorporating block-wise quantization notably enhances visual quality under INT4, improving from 16.79 to 46.53. Further integrating reorder to balance the data distribution within blocks further elevates the generation quality. Ultimately, by adopting mixed precision quantization, "PARO-MP" achieves algorithm performance on par with INT8 and FP16, averaging 4.80bit.

Analysis of Attention Patterns. We visualize the attention maps before and after reorder in Fig. 8. The reorder effectively unifies diverse patterns into quantization- and hardware-friendly "block diagonal" pattern. It highlights that different attention heads conduct local aggregation along various dimensions, such as "frame" and "height" in the examples shown.

End-to-end Speedup. As shown in Fig. 6(a), we normalize the performance of PARO, other accelerators and GPUs to Sanger. Compared with Sanger and ViTCoD, PARO obtains performance improvements of 10.61/12.04× and 6.38/7.05× on CogVideoX-2B/5B, respectively. Although the A100 achieves higher end-to-end performance than PARO, this is mainly due to the higher peak performance and memory bandwidth of A100. When PARO uses the same hardware parameters as the A100, it achieves a performance improvement of 1.68/2.71× compared to the A100. This is primarily due to the mixed-precision attention quantization and our output-bitwidth aware mixed-precision computation flow.

Ablation Study for Performance. The breakdown of PARO's performance gains are illustrated in Fig. 6(b). The naive FP16 version includes no optimizations in PARO. Building on this baseline, we introduced W8A8 linear layer quantization, 4.80bit attention layer quantization, and output-bitwidth aware mixed-precision computing unit. For the 2B/5B models, linear layer quantization achieved a 1.07/1.11× speedup. Adding attention mixed-precision quantization further improved the speedup to 2.33/2.38×. Finally, incorporating the output-bitwidth aware mixed-precision optimization increased the speedup to 3.06/3.00×. Among these optimizations, 4.80bit attention quantization contributed the most significant improvement, as 3D full attention is the performance bottleneck.

Reorder Overhead. We evaluate the overhead of introducing the reorder for $QKVO$ during inference in PARO. Experimental results show that for the CogVideoX-2B/5B models, the reorder operation accounts for only 1.26% and 1.07% of the end-to-end latency, respectively. This is because the data size of the $QKVO$ matrices is only 0.36% of the attention map, making the overhead negligible in the highly compute-bound attention computations.

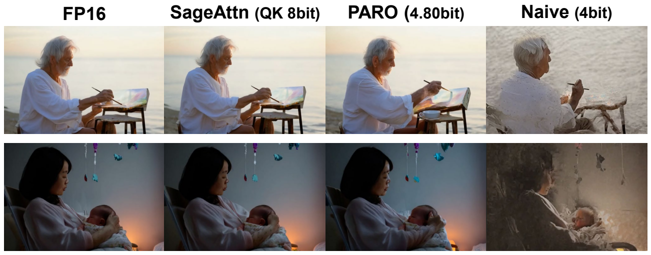

Fig. 7. Comparison of generated videos for different quantization methods. The PARO quantization method could generate videos without visual difference with FP16 generated videos with an averaged of 4.80bit.

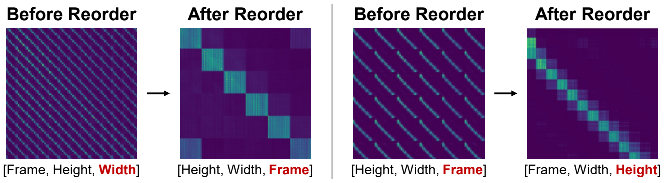

Fig. 8. Visualization of attention pattern before and after reorder. The reorder unifies different patterns into "block diagonal" pattern.

Energy Efficiency. Thanks to our hardware-friendly attention quantization and efficiently mixed-precision PE array, PARO achieves energy efficiency of 3.46/3.61 TOPS/W on the CogVideoX-2B/5B models, respectively, which are 4.86/6.43× higher compared to NVIDIA A100 GPU.

VI. Conclusions

We present PARO, a video generation accelerator that leverages hardware-software co-design with pattern-aware reorder-based attention quantization. PARO examines 3D full attention patterns and introduces a reorder technique to consolidate them into a block diagonal structure. Block-wise mixed precision quantization achieves lossless compression with an average bitwidth of 4.80. The hardware integrates a dynamically reconfigurable mixed-precision PE array, ensuring full utilization of mixed-precision characteristics. Experimental results demonstrate PARO delivers up to 2.71× improvement in end-to-end performance compared to an NVIDIA A100 GPU and achieves up to 6.38∼7.05× speedup over state-of-the-art ASIC-based accelerators on the CogVideoX-2B and 5B models.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 62325405, 62104128, U21B2031, 62204164), the National Key R&D Program of China (2023YFB4502200), Tsinghua EE Xilinx AI Research Fund, Tsinghua-Meituan Joint Institute for Digital Life, and Beijing National Research Center for Information Science and Technology (BNRist).

References

[1] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4195–4205.

[2] HPC-AI, "Open-Sora," https://github.com/hpcaitech/Open-Sora, 2024.

[3] OpenAI, "Video generation models as world simulators," https://openai.com/index/video-generation-models-as-world-simulators/, 2024.

[4] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng *et al.*, "Cogvideox: Text-to-video diffusion models with an expert transformer," *arXiv preprint arXiv:2408.06072*, 2024.

[5] X. Huang, Z. Shen, and K.-T. Cheng, "Variation-aware vision transformer quantization," *arXiv e-prints*, pp. arXiv–2307, 2023.

[6] Z. Yuan, C. Xue, Y. Chen, Q. Wu, and G. Sun, "Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization," in *European conference on computer vision*. Springer, 2022, pp. 191–207.

[7] Z. Li and Q. Gu, "I-vit: Integer-only quantization for efficient vision transformer inference," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 065–17 075.

[8] Y. Li, S. Xu, B. Zhang, X. Cao, P. Gao, and G. Guo, "Q-vit: Accurate and fully quantized low-bit vision transformer," *Advances in neural information processing systems*, vol. 35, pp. 34 451–34 463, 2022.

[9] Z. Li, J. Xiao, L. Yang, and Q. Gu, "Repq-vit: Scale reparameterization for post-training quantization of vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 227–17 236.

[10] Y. Lin, T. Zhang, P. Sun, Z. Li, and S. Zhou, "Fq-vit: Post-training quantization for fully quantized vision transformer," *arXiv preprint arXiv:2111.13824*, 2021.

[11] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[12] P. Dong, L. Lu, C. Wu, C. Lyu, G. Yuan, H. Tang, and Y. Wang, "Packqvit: Faster sub-8-bit vision transformers via full and packed quantization on the mobile," *Advances in Neural Information Processing Systems*, vol. 36, pp. 9015–9028, 2023.

[13] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2704–2713.

[14] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. Van Baalen, and T. Blankevoort, "A white paper on neural network quantization," *arXiv preprint arXiv:2106.08295*, 2021.

[15] T. Zhao, T. Fang, E. Liu, R. Wan, W. Soedarmadji, S. Li, Z. Lin, G. Dai, S. Yan, H. Yang *et al.*, "Vidit-q: Efficient and accurate quantization of diffusion transformers for image and video generation," *arXiv preprint arXiv:2406.02540*, 2024.

[16] J. Zhang, J. Wei, P. Zhang, J. Zhu, and J. Chen, "Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration," in *International Conference on Learning Representations (ICLR)*, 2025.

[17] J. Zhang, H. Huang, P. Zhang, J. Wei, J. Zhu, and J. Chen, "Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization," 2024. [Online]. Available: https://arxiv.org/abs/2411.10958

[18] J. Zhang, C. Xiang, H. Huang, H. Xi, J. Wei, J. Zhu, and J. Chen, "Spargeattn: Accurate sparse attention accelerating any model inference," 2025.

[19] T. J. Ham, S. J. Jung, S. Kim, Y. H. Oh, Y. Park, Y. Song, J.-H. Park, S. Lee, K. Park, J. W. Lee *et al.*, "A^ 3: Accelerating attention mechanisms in neural networks with approximation," in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2020, pp. 328–341.

[20] H. Wang, Z. Zhang, and S. Han, "Spatten: Efficient sparse attention architecture with cascade token and head pruning," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2021, pp. 97–110.

[21] T. J. Ham, Y. Lee, S. H. Seo, S. Kim, H. Choi, S. J. Jung, and J. W. Lee, "Elsa: Hardware-software co-design for efficient, lightweight self-attention mechanism in neural networks," in *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2021, pp. 692–705.

[22] Z. Qu, L. Liu, F. Tu, Z. Chen, Y. Ding, and Y. Xie, "Dota: detect and omit weak attentions for scalable transformer acceleration," in *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2022, pp. 14–26.

[23] H. Fan, T. Chau, S. I. Venieris, R. Lee, A. Kouris, W. Luk, N. D. Lane, and M. S. Abdelfattah, "Adaptable butterfly accelerator for attention-based nns via hardware and algorithm co-design," in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2022, pp. 599–615.

[24] P. Dong, M. Sun, A. Lu, Y. Xie, K. Liu, Z. Kong, X. Meng, Z. Li, X. Lin, Z. Fang *et al.*, "Heatvit: Hardware-efficient adaptive token pruning for vision transformers," in *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2023, pp. 442–455.

[25] L. Lu, Y. Jin, H. Bi, Z. Luo, P. Li, T. Wang, and Y. Liang, "Sanger: A co-design framework for enabling sparse attention using reconfigurable architecture," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, 2021, pp. 977–991.

[26] H. You, Z. Sun, H. Shi, Z. Yu, Y. Zhao, Y. Zhang, C. Li, B. Li, and Y. Lin, "Vitcod: Vision transformer acceleration via dedicated algorithm and accelerator co-design," in *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2023, pp. 273–286.

[27] Z. Song, C. Qi, F. Liu, N. Jing, and X. Liang, "Cmc: Video transformer acceleration via codec assisted matrix condensing," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, 2024, pp. 201–215.

[28] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[29] T. Unterthiner, S. Van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Fvd: A new metric for video generation," 2019.

[30] C. Wu, L. Huang, Q. Zhang, B. Li, L. Ji, F. Yang, G. Sapiro, and N. Duan, "Godiva: Generating open-domain videos from natural descriptions," *arXiv preprint arXiv:2104.14806*, 2021.

[31] P. Esser, J. Chiu, P. Atighehchian, J. Granskog, and A. Germanidis, "Structure and content-guided video synthesis with diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7346–7356.

[32] H. Wu, E. Zhang, L. Liao, C. Chen, J. Hou, A. Wang, W. Sun, Q. Yan, and W. Lin, "Exploring video quality assessment on user generated contents from aesthetic and technical perspectives," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 144–20 154.

[33] R. Balasubramonian, A. B. Kahng, N. Muralimanohar, A. Shafiee, and V. Srinivas, "Cacti 7: New tools for interconnect exploration in innovative off-chip memories," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 14, no. 2, pp. 1–25, 2017.