RainFusion2.0: Temporal-Spatial Awareness and Hardware-Efficient Block-wise Sparse Attention

Aiyue Chen[*1] chenaiyue@huawei.com

Yaofu Liu[*1,2] yliuls@connect.ust.hk

Junjian Huang[*1] huangjunjian3@huawei.com

Guang Lian[*1] lianguang@huawei.com

Yiwu Yao[*1] yiwuyao@pku.edu.cn

Wangli Lan[*1] lanwangli@huawei.com

Jing Lin[*1] linjing28@huawei.com

Zhixin Ma[*1] mazhixin7@huawei.com

Tingting Zhou[*1] zhoutingting13@huawei.com

Harry Yang[2†]

Abstract

In video and image generation tasks, Diffusion Transformer (DiT) models incur extremely high computational costs due to attention mechanisms, which limits their practical applications. Furthermore, with hardware advancements, a wide range of devices besides graphics processing unit (GPU), such as application-specific integrated circuit (ASIC), have been increasingly adopted for model inference. Sparse attention, which leverages the inherent sparsity of attention by skipping computations for insignificant tokens, is an effective approach to mitigate computational costs. However, existing sparse attention methods have two critical limitations: the overhead of sparse pattern prediction and ***the lack of hardware generality****, as most of these methods are designed for GPU. To address these challenges, this study proposes RainFusion2.0, which aims to develop an online adaptive, hardware-efficient, and low-overhead sparse attention mechanism to accelerate both video and image generative models, with robust performance across diverse hardware platforms.* ***Key technical insights*** *include: (1) leveraging block-wise mean values as representative tokens for sparse mask prediction; (2) implementing spatiotemporal-aware token permutation; and (3) introducing a first-frame sink mechanism specifically designed for video generation scenarios. Experimental results demonstrate that RainFusion2.0 can achieve 80% sparsity while achieving an end-to-end speedup of 1.5 1.8× without compromising video quality. Moreover, RainFusion2.0 demonstrates effectiveness across various generative models and validates its* ***generalization across diverse hardware platforms****.*

1. Introduction

Diffusion Transformer (DiT) models have exhibited outstanding performance in visual generation tasks. However, their tremendously high computational cost limits the practical application of DiT models. Among these costs, the major component stems from the Attention mechanism, whose computation scales exponentially with token length—ranging from 10K to 80K in current state-of-the-art DiT models. As research advances, it has been revealed that the Attention mechanism possesses inherent sparsity: the softmax operation in Attention enables a minor subset of tokens to exert a dominant impact on the final output. Consequently, sparse Attention, which accelerates Attention computation by only focusing on critical tokens, has emerged as a prevalent optimization approach. Currently, sparse Attention methods can be primarily categorized into three types: (1) fixed patterns, (2) online adaptive masks, and (3) token permutations. Nevertheless, most of these existing methods suffer from certain limitations, which restrict their widespread practical adoption.

Limitations: 1.The Dilemma of balancing Accuracy and Efficiency. Two main factors contribute to this dilemma: (1) sparsity degree; (2) the cost of predicting sparse patterns. For fixed patterns, the overhead of prediction is relatively small. However, to achieve high accuracy, fixed-pattern methods typically require a low sparsity degree, which consequently compromises efficiency. On the other hand, online adaptive mask-based and token permutation-based methods involve substantial overhead, primarily due to the

[*] All authors contribute equally to this work as the first authors.

[†] Corresponding author: Harry Yang. E-mail: yangharry@ust.hk

[1] Huawei Technologies Co., Ltd [2] The Hong Kong University of Science and Technology

costs associated with predicting the sparse mask and the token permutation layout. 2.Lack of Device Universality. A major shortcoming of existing works is their neglect of universality across various devices. Most prediction methods are specifically designed for Graphics Processing Units (GPUs) and are not universally applicable to other Artificial Intelligence (AI) devices such as Application Specific Integrated Circuit(ASIC). Specifically, the prediction process incurs unacceptably long time costs on AI devices other than GPUs.

Goal: We aim to design an online adaptive, hardware-efficient, and overhead-free sparse attention mechanism to accelerate video and image generation models.

Key Idea: 1.Adaptive and Efficient Design. We partition tokens into different blocks and use the mean value of each block as the representative token for predicting the sparse mask. 2.Spatiotemporal-aware permutation. To enhance the similarity within each block, we adopt a spatiotemporal-aware permutation strategy. Based on the inherent spatial relationships of videos or images, tokens are divided into three-dimensional (3D) or two-dimensional (2D) windows. Tokens within each window are arranged adjacently and then flattened window by window. 3.First Frame Sink for Video Generation. For video generation models, we fix the attention computation relationship with the tokens of the first frame. This fixed pattern is termed "First Frame Sink". 4.Hardware Generality. Most importantly, our proposed method not only features low overhead but also achieves generality across vary types of AI devices.

2. Related Work

Sparse attention mechanisms have garnered significant research interest within the domain of video and image generative diffusion models. This work categorizes these advancements into three primary paradigms:

(1) **Fixed Sparse Patterns**: Sparse attention with fixed patterns typically employs a predetermined set of sparse masks. The model selects one of these masks for use. For instance, Sliding Tile Attention (STA)[7] precomputes a set of sliding window masks. Leveraging a small number of prompts, STA matches an optimal pattern for each layer and each head within the Denoising Diffusion Transformer (DiT). During the generation of new videos, these pre-matched masks are directly used. Similarly, Sparse Video Generation (SVG)[3] designs two distinct pattern types: temporal and spatial masks. When computing attention, SVG dynamically checks which mask is most suitable for the current attention head. Rainfusion[1] extends this concept by introducing three mask types: temporal, spatial, and textural. Analogous to SVG, Rainfusion dynamically assigns the most appropriate mask to each attention head.

(2) **Online Adaptive Patterns**: This category of sparse attention does not rely on fixed masks. Instead, tokens are generally partitioned into blocks, and the model determines online which block tokens will participate in the attention computation. SparseAttention [6] calculates the mean for each block and uses these block means to derive attention scores. It then selects the top-k blocks such that their cumulative distribution function (CDF) meets a specific threshold. Concurrently, it computes the cosine similarity between tokens within each block. Adaptive Sparse Attention (AdaSpa)[4] observes that the sparse patterns for each layer and head in DiT models remain stable throughout the diffusion steps. Consequently, in the initial timesteps, it maintains full attention and online computes the optimal sparse pattern for each layer, which is then reused in subsequent steps.

(3) **Token Permutations**: This approach focuses on rearranging similar tokens to be adjacent, thereby enabling effective sparsification. PAROAttention[8] primarily explores permuting video tokens across different dimensions, such as frame, height, and width, resulting in six possible orderings. Sparse Video Generation 2 (SVG2)[5] employs K-means clustering to group similar tokens together. It then uses the centroid of each cluster as its representative. Based on these centroids, SVG2 computes the attention scores between clusters to determine which inter-cluster attention computations can be strategically omitted.

3. Method

3.1. Blockwise Sparse

Flash attention is the backbone of the modern transformer model. RainFusion2 adopts flash attention into the sparse mode. Consider the attention operation: given the query, key, value matrices $Q, K, V \in \mathbb{R}^{N\times d}$, we have $S = QK^\top/\sqrt{d}, \quad P = \text{Softmax}(S), \quad O = PV$,

where the Softmax function is defined element-wise as $\text{Softmax}(S)_{ij} = \frac{\exp(S_{ij})}{\sum_k \exp(S_{ik})}$.

Flash attention: Divide Q into T_q blocks $\{Q_i\}$ with block size b_q, where $Q_i \in \mathbb{R}^{b_q\times d}$ and $T_q = [N/b_q]$. Divide K and V into T_k blocks $\{K_i\}$ and $\{V_i\}$ respectively, with the same block size b_k, where $K_i \in \mathbb{R}^{b_k\times d}$, $V_i \in \mathbb{R}^{b_k\times d}$ and $T_k = [N/b_k]$.

Subsequently, it use online softmax to compute each block of the output O iteratively. The calculation of block O_i proceeds as:

$$S_{i,j} = Q_i K_j^\top/\sqrt{d}, \tag{1}$$

$$(m_{i,j}, \tilde{P}_{i,j}) = \tilde{f}(m_{i,j-1}, S_{i,j}), \tag{2}$$

$$l_{i,j} = \exp\left(m_{i,j-1} - m_{i,j}\right) l_{i,j-1} + \text{rowsum}(\tilde{P}_{i,j}), \tag{3}$$

$$O_{i,j} = \text{diag}\left(\exp\left(m_{i,j-1} - m_{i,j}\right)\right) O_{i,j-1} + \tilde{P}_{i,j} V_j \tag{4}$$

where $m_{i,j}, l_{i,j} \in \mathbb{R}^{b_q\times 1}$, with initial values set to $-\infty$ and

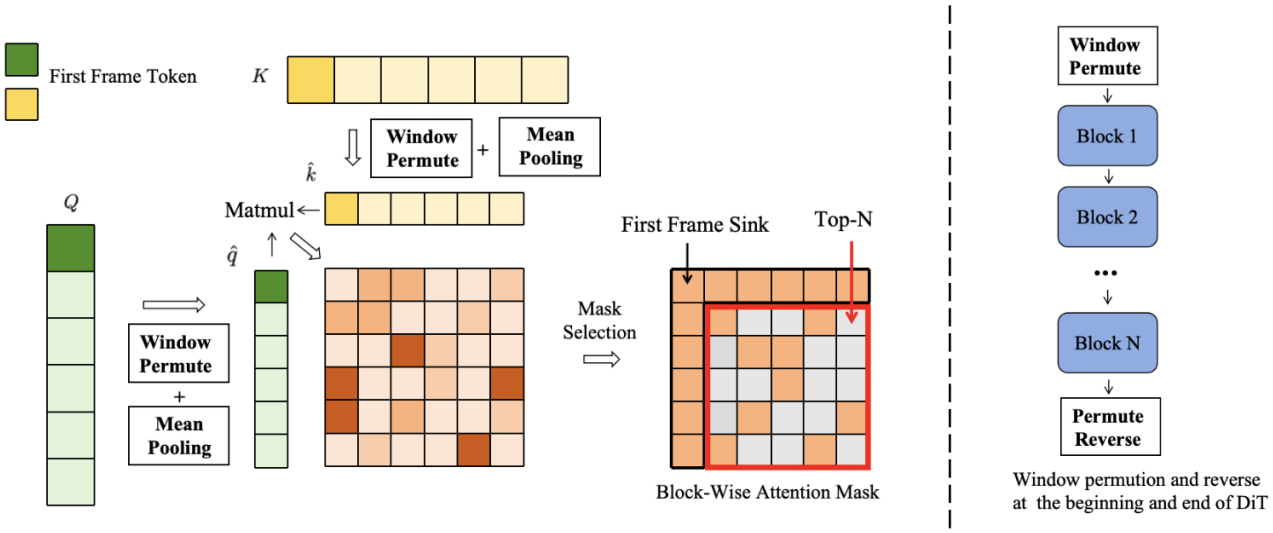

Figure 1. Workflow of RainFusion2.0

0 respectively. The operator $\tilde{f}(\cdot)$ is defined as follows: it computes $m_{i,j} = \max\{m_{i,j-1}, \text{rowmax}(S_{i,j})\}$ and $\tilde{P}_{i,j} = \exp(S_{i,j} - m_{i,j})$. Finally, the block O_i (the final output of this incremental process) is obtained via:

$$O_i = \text{diag}(l_{i,j})^{-1} O_{i,j}$$

To accelerate attention computation and improve hardware utilization, we either skip or compute the full block-wise matrix multiplications of $Q_i K_j^\top$ and $P_{i,j} V_j$. We define a block-wise attention mask M with dimensions $\lceil N/b_q \rceil \times \lceil N/b_k \rceil$, where each element takes a value of either 0 or 1. The sparse Flash Attention mechanism is then implemented as follows:

$$Q_i K_j^\top \text{ and } P_{i,j} V_j \text{ are skipped if } M_{ij} = 0.$$

3.2. Representative Token for Blocks

To derive the block-wise attention mask M, if we calculate the full attention scores P as $Q \times K^\top$, this approach actually offers no sparsity-induced speedup. Thus, we select representative tokens $\hat{q}_i$ and $\hat{k}_i$ for each block Q_i (query block) and K_i (key block), respectively.

The core idea underlying the design of representative tokens originates from the observation that adjacent tokens in the Q, K matrix exhibit high similarity—a characteristic that is consistent across different models and also across distinct attention layers within a single model.

The representative tokens $\hat{q}_i$ and $\hat{k}_j$ are defined as follows:

$$\hat{q} = \{\hat{q}_i\} = \text{mean}(Q_i, \text{axis} = 0), \tag{5}$$

$$\hat{k} = \{\hat{k}_j\} = \text{mean}(K_j, \text{axis} = 0), \tag{6}$$

$$\hat{S} = \{\hat{S}_{ij}\} = \{\hat{q}_i \hat{k}_j^\top\} \tag{7}$$

where the matrix $\hat{S}$ has dimensions $\lceil N/b_q \rceil \times \lceil N/b_k \rceil$, and $\{\hat{S}_{ij}\}$ serves as an indicator to quantify the contribution of the block pair (Q_i, K_j) to the overall attention scores. Subsequently, we select the top-n important blocks $\{K_j\}$ for each block $\{Q_i\}$ based on the values of $\hat{S}_{ij}$.

$$M_{ij} = \begin{cases} 1, & \text{if } j \in \left\{ j \mid \hat{S}_{ij} \in \text{TopN}\left(\hat{S}, \text{dim} = 0\right) \right\}, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

However, a key challenge persists in this design: Despite the high similarity between adjacent tokens, those within a single block still show considerable variability. To mitigate this issue, some studies compute the cosine similarity of tokens inside blocks Q_i and K_j; yet, this approach imposes considerable computational overhead. In contrast, we employ window permutation techniques to enhance the similarity of tokens within blocks Q_i and K_j.

3.3. 3D Window Permutation

In video generation diffusion models, although tokens are flattened into 1D sequences, they inherently encode 3D physical spatiotemporal information. The length of the token sequence is given by $N = F \cdot W \cdot H$, where F, H, and W denote the number of frames, spatial height, and spatial width of the video in the latent space, respectively. The tokens typically exhibit similarity to adjacent tokens in both temporal and spatial dimensions.

The default token layout in diffusion models is $[F, H, W]$. However, this order weakens the adjacent similarity in both the temporal and spatial dimensions. Tokens that are adjacent in 3D space are shuffled into different 1D positions. Consequently, the tokens within each block

are essentially non-adjacent in 3D space, thus undermining their self-similarity.

Permutation: To enhance the similarity of tokens within each block (i.e., Q_i and K_j), we rearrange the tokens in a 3D window-based order. The details will be released later.

3.4. First Frame Sink

Similar to the well-documented "attention sink" phenomenon in large language models (LLMs), where initial tokens receive high levels of attention, an analogous phenomenon exists in video generation models with 3D full attention. We refer to this analogous phenomenon in video generation as the *First Frame Sink*.

Our observations across a range of video generation models reveal that the first frame exerts a substantial influence on the quality of the final generated video. Specifically, omitting the attention computation involving tokens corresponding to the first frame leads to a non-trivial degradation in the quality of the generated video. Thus, tokens in Q that represent the first frame are designed to attend to all tokens in K, while all tokens in Q are enforced to attend to tokens in K that correspond to the first frame.

In the Fig. 1, the first frame token is shown at the beginning of the sequence. However, in practice, we move the first frame token to the end of the sequence. This is because some models combine video and text tokens as multimodal input for attention computation. By grouping the first frame token with the text tokens, we can ensure they both participate in full attention.

4. Experiment

4.1. setup

Device: We evaluated our method on the Neural Processing Unit (NPU), a type of Application-Specific Integrated Circuit (ASIC). As a typical and prevalent AI device distinct from Graphics Processing Units (GPUs), NPUs have been widely deployed for the inference of diffusion models. To highlight the hardware universality and efficiency of our proposed method, we conducted comprehensive performance evaluations on NPUs.

Model: We validated the effectiveness of RainFusion2 by applying it to various generative models, covering both video and image generation tasks.

Video Generation Models: Experiments were conducted on Wan2.2 and Hunyuanvideo1.5. For Wan2.2, we performed tests on the 720P resolution. For Hunyuanvideo1.5, evaluations were carried out on both 480P and 720P video resolutions. Notably, Hunyuanvideo1.5 inherently incorporates a sparse attention mechanism when generating 720P videos. We thus conducted comparative experiments between RainFusion2 and this native sparse attention mechanism in terms of both accuracy and performance.

Image Generation Model: We additionally implemented experiments on the image generation model Qwen-image-edit.

Evaluation Metrics: For video generation models, Vbench[2] was primarily adopted as the core evaluation metric for video quality, which encompasses multiple critical dimensions including subjective consistency, imaging quality, and overall consistency. Furthermore, we introduced cosine similarity to characterize the overall differences between videos generated by the full attention mechanism and those generated by RainFusion2.

Baseline: Full Attention was employed as the fundamental baseline for comparison. We tested RainFusion2 with different sparsity levels (80% and 90%). Other sparse attention methods (e.g., SparseAttention and SVG2) exhibit poor compatibility with NPUs, and thus were not included in the efficiency comparison experiments.

Perceptual Evaluation: While Vbench metrics can reflect the overall quality of videos and cosine similarity can characterize overall differences, these metrics struggle to capture the fine-grained differences that are perceptible to the human eye. To address this limitation, we incorporated perceptual evaluation to quantitatively assess the detailed differences between videos.

4.2. Main result

Video Generation Models: For Wan2.2 generating 720p videos, we achieve a speedup of 1.57× to 1.8× with a sparsity ratio of 80% to 90%, while maintaining nearly imperceptible quality loss (visually indistinguishable from the full attention baseline). For HunyuanVideo1.5, configuring a sparsity ratio of 80% to 90% also yields videos with almost no visual quality degradation. Specifically, at 80% sparsity, HunyuanVideo1.5 achieves a 1.16× speedup for 480p video generation and a 1.28× speedup under the 720p setting.

Image Generation Model: We also validated our method on the Qwen Image Edit model. Even with a sparsity ratio of 60%, the model maintains consistent quality in generated images (visually aligned with the full attention baseline).

Ablation Study: We conduct ablation experiments on the 3D window permutation. Although omitting permutation still performs well in quality metrics, and its cosine similarity is nearly identical to that of the videos generated with full attention, visual inspection of the generated videos reveals minor yet noticeable differences in temporal motion smoothness and frame details. For example, as shown in Fig. 3, videos generated without 3D window permutation are generally similar to those of full attention, but obvious flaws appear in some local areas — these differences are hard to capture with common quality metrics. In contrast, adding 3D window permutation results in videos that are visually consistent with full attention, both in overall ap-

Table 1. RainFusion2 and Full Attention Comparison: Quality & Efficiency Metrics (Wan2.2 720p)

Method(sparsity)	Quality Metrics				Efficiency Metrics	
	Subj. Cons. ↑	Imaging Qual. ↑	Overall Cons. ↑	Cosine Sim. ↑	Latency (s)	Speedup
Full Attention	0.9717	0.6816	0.2591		532	
RainFusion(80%)(w/o 3D order)	0.9690	0.6791	0.2555	0.9532	339	1.57x
RainFusion(90%)(w/o 3D order)	0.9643	0.6709	0.2555	0.9476	295	1.80x
RainFusion(80%)(w/ 3D order)	0.9683	0.6864	0.2562	0.9514	339	1.57x

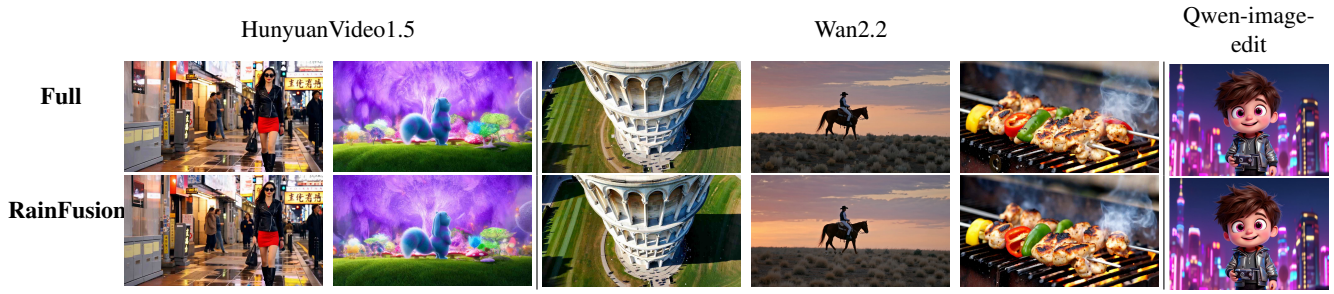

Figure 2. Results of RainFusion on Diffusion Models. HunyuanVideo1.5 and Wan2.2 generate 720p videos under two configurations: full attention and RainFusion with 80% sparsity. Qwen-image-edit generates 1024×1024 images using RainFusion with 60% sparsity.

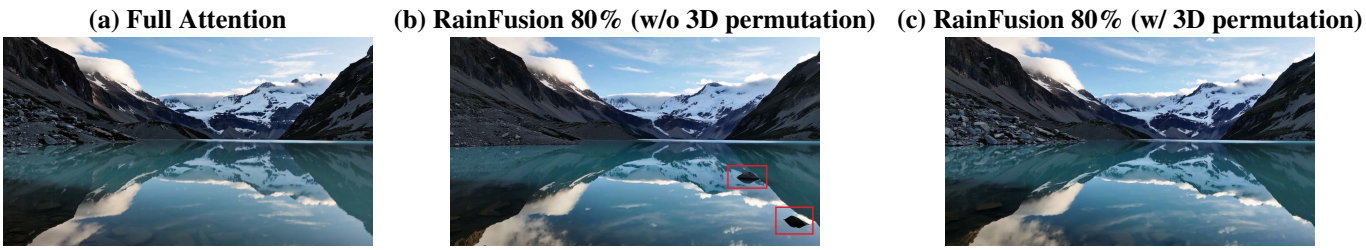

Figure 3. Experimental results on the Wan2.2 dataset. As shown in Subfigure (b), the video generated by RainFusion (80% sparsity, without 3D permutation) is overall comparable to that of full attention (Subfigure (a)). However, two prominent spurious rock artifacts emerge in the bottom-right corner of the video frame. In contrast, these visual artifacts are completely eliminated when 3D permutation is integrated into RainFusion (Subfigure (c)).

pearance and fine-grained details.

5. Conclusion

In conclusion, this study has addressed the critical challenges of high computational cost in DiT and limited hardware generality in sparse attention by proposing RainFusion2.0. Through the innovative integration of block-wise representative token prediction, spatiotemporal-aware token permutation, and a first-frame sink mechanism, RainFusion2.0 achieves a remarkable balance between efficiency and performance. The experimental results demonstrate that it can achieve 80% sparsity, leading to an end-to-end speedup of 1.5 1.8× across various video and image generative models, all while maintaining high output quality. More importantly, its hardware-efficient design allows RainFusion2.0 to work across a wide range of platforms, effectively mitigating the hardware-specific limitations of existing sparse attention methods. This work not only provides a practical and high-performance solution for accelerating DiT models but also paves the way for their broader deployment on heterogeneous computing devices. Future work will focus on exploring how to combine RainFusion2.0 with other orthogonal acceleration methods, such as quantization and distillation.

References

[1] Aiyue Chen et al. Rainfusion: Adaptive video generation acceleration via multi-dimensional visual redundancy. In arXiv preprint arXiv:2505.21036, 2025. 2

[2] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. In CVPR, pages 21807–21818, 2024. 4

[3] Haocheng Xi et al. Sparse videogen: Accelerating video

diffusion transformers with spatial-temporal sparsity. In Forty-second International Conference on Machine Learning, 2025. 2

[4] Yifei Xia, Suhan Ling, Fangcheng Fu, Yujie Wang, Huixia Li, Xuefeng Xiao, and Bin Cui. Training-free and adaptive sparse attention for efficient long video generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 15982–15993, 2025. 2

[5] Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, Jianfei Chen, Song Han, Kurt Keutzer, and Ion Stoica. Sparse videogen2: Accelerate video generation with sparse attention via semantic-aware permutation. In The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025. 2

[6] Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattention: Accurate and training-free sparse attention accelerating any model inference. In Forty-second International Conference on Machine Learning, 2025. 2

[7] Peiyuan Zhang et al. Fast video generation with sliding tile attention. In Forty-second International Conference on Machine Learning, 2025. 2

[8] Tianchen Zhao, Ke Hong, Xinhao Yang, Xuefeng Xiao, Huixia Li, Feng Ling, Ruiqi Xie, SiQi Chen, Hongyu Zhu, Zhang Yichong, and Yu Wang. PAROAttention: Pattern-aware reordering for efficient sparse and quantized attention in visual generation models. In The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025. 2