

1

FPSAttention Supplementary Material

2

Contents

3	A Implementation Details	2
4	A.1 FPSAttention Algorithm	2
5	A.2 Additional Implementation Details	3
6	A.3 Ablation Study: Challenges of Naive Quantization and Sparsity Combination . . .	3
7	A.4 VBench Full Evaluation Results	5
8	A.5 Clarification on vBench Evaluation Metrics	5
9	A.6 Training Hyperparameters	5
10	B Limitation and Societal Impacts	6
11	C Visualization	8

13 A Implementation Details

14 A.1 FPSAttention Algorithm

15 Algorithm 1 presents the core computational workflow of our FPSAttention method, which imple-
 16 ments joint tile-wise FP8 quantization with structured sparse attention and denoising step-aware
 17 adaptation. The algorithm follows the methodology described in the main paper, incorporating tile-
 18 wise quantization for queries and keys, channel-wise quantization for values, tensor-wise quantization
 19 for attention weights, and dynamic adaptation based on denoising timesteps.

Algorithm 1 FPSAttention: Joint Tile-wise FP8 Quantization and Sparse Attention

Require: Input tensors $Q, K, V \in \mathbb{R}^{L \times d}$, denoising step t , diffusion steps D
Require: Transition points α_1, α_2 , quantization granularities $\{g_{\text{coarse}}, g_{\text{fine}}, g_{\text{intermediate}}\}$
Require: Window sizes $\{W_{\text{sparse}}, W_{\text{dense}}, W_{\text{medium_density}}\}$, tile scheme \mathcal{T}
Ensure: Output tensor $X \in \mathbb{R}^{L \times d}$ (BF16/FP16)

- 1: // 1. Denoising Step-aware Parameter Selection
- 2: $t_1 \leftarrow \alpha_1 \cdot D, t_2 \leftarrow \alpha_2 \cdot D$
- 3: **if** $t \leq t_1$ **then**
- 4: $g(t) \leftarrow g_{\text{coarse}}, W(t) \leftarrow W_{\text{sparse}}$ // Early steps
- 5: **else if** $t_1 < t \leq t_2$ **then**
- 6: $g(t) \leftarrow g_{\text{fine}}, W(t) \leftarrow W_{\text{dense}}$ // Mid steps
- 7: **else**
- 8: $g(t) \leftarrow g_{\text{intermediate}}, W(t) \leftarrow W_{\text{medium_density}}$ // Late steps
- 9: **end if**
- 10: // 2. Tile-wise FP8 Quantization for Q and K
- 11: Partition Q, K into tiles $\{\mathcal{T}_u\}$ with granularity $g(t)$
- 12: **for** each tile \mathcal{T}_u **do**
- 13: $s_u^Q \leftarrow \max_{(i,j) \in \mathcal{T}_u} |Q_{i,j}| / M_{\text{FP8_max}}$
- 14: $s_u^K \leftarrow \max_{(i,j) \in \mathcal{T}_u} |K_{i,j}| / M_{\text{FP8_max}}$
- 15: $\hat{Q}_{i,j} \leftarrow \text{FP8}(Q_{i,j}; s_u^Q)$ for $(i, j) \in \mathcal{T}_u$
- 16: $\hat{K}_{i,j} \leftarrow \text{FP8}(K_{i,j}; s_u^K)$ for $(i, j) \in \mathcal{T}_u$
- 17: **end for**
- 18: // 3. Channel-wise FP8 Quantization for V
- 19: **for** each channel $j \in \{1, \dots, d\}$ **do**
- 20: $s_j^V \leftarrow \max_{i \in L} |V_{i,j}| / M_{\text{FP8_max}}$
- 21: $\hat{V}_{i,j} \leftarrow \text{FP8}(V_{i,j}; s_j^V)$ for all i
- 22: **end for**
- 23: // 4. Structured Sparse Attention Computation via FlexAttention
- 24: Define neighborhood $\mathcal{W}(u)$ based on window size $W(t)$:
- 25: $\mathcal{W}(u) = \{v : \|c_u - c_v\|_\infty \leq (W_t/(2T_t), W_h/(2T_h), W_w/(2T_w))\}$
- 26: // Configure FlexAttention mask and score modification functions
- 27: Define $\text{mask_mod}(b, h, q, k) = \text{True}$ if $\text{tile}(q) \in \mathcal{W}(\text{tile}(k))$, **False** otherwise
- 28: Define $\text{score_mod}(S, b, h, q, k) = S$ // Identity for quantized inputs
- 29: // Execute FlexAttention with quantized inputs and custom modifications
- 30: $\hat{X} \leftarrow \text{FlexAttention}(\hat{Q}, \hat{K}, \hat{V}, \text{score_mod}, \text{mask_mod})$
- 31: // 5. Dequantize Output to Target Precision
- 32: $X \leftarrow \text{Dequantize}(\hat{X})$ // Dequantize to BF16/FP16
- return** X

20 Key Algorithmic Components

21 The algorithm implements the four core innovations described in the main paper:

- **Denoising Step-aware Adaptation:** Lines 2-8 implement the adaptive scheduling strategy from Equation 6 in the main paper, dynamically adjusting quantization granularity $g(t)$ and sparsity window size $W(t)$ based on the current denoising step t .
- **Tile-wise FP8 Quantization for Q and K:** Lines 10-15 partition queries and keys into 3D tiles with step-dependent granularity and compute per-tile scaling factors s_u^Q and s_u^K to minimize quantization error within each tile.
- **Channel-wise FP8 Quantization for V:** Lines 17-20 apply channel-wise quantization to the value matrix, preserving fine-grained channel information that is critical for generation quality.
- **FlexAttention-based Sparse Attention:** Lines 22-26 implement structured sparse attention using FlexAttention’s `mask_mod` and `score_mod` interfaces, enabling hardware-optimized execution with tile-wise sparsity patterns that generate exactly $M \times |\mathcal{W}(u)|$ dense attention blocks.
- **Output Dequantization:** Line 28 dequantizes the FlexAttention output to the target precision (BF16/FP16) to maintain compatibility with the downstream network components.

This implementation ensures full compatibility with the theoretical framework while enabling practical hardware acceleration through structured computation patterns and optimal memory access patterns.

A.2 Additional Implementation Details

Models. We implement and evaluate FPSAttention on the Wan architecture [5], leveraging both 1.3B and 13B parameter variants to demonstrate scalability. The Wan models feature a DiT backbone with cross-attention for text conditioning and temporal attention for inter-frame modeling. Our implementation maintains architectural fidelity while seamlessly integrating FP8 quantization for attention. Especially, we used the E4M3 formats for FP8 quantization for attention. The joint quantization and sparsity mechanisms are realized through FlexAttention’s score and mask modification interfaces, with the resulting fused kernels compiled via Triton for optimal execution on Hopper architectures.

Hardware. Experiments utilize a distributed computing cluster with high-performance GPU nodes, each containing 192 CPU cores, 960GB system memory, and 8×NVIDIA H20 GPUs (96GB each). InfiniBand interconnects ensure high-bandwidth inter-node communication for distributed training. Training scales from 16 nodes (1.3B model) to 64 nodes (13B model), requiring approximately 7 days. We trained around 6000 iterations per configuration to achieve convergence.

Dataset. Training employs a curated high-quality video dataset processed through a comprehensive filtering pipeline. The preprocessing workflow includes automated subtitle removal, black-border cropping, and monochrome video exclusion, followed by quality-based filtering using established metrics (Q-Align > 3.5, Aesthetic Score > 2.0, optical flow magnitude 0.05–2.0). After deduplication, videos are standardized to 480p resolution, 16fps frame rate, and 5-second duration to optimize the computational efficiency-quality balance across both model scales.

Evaluation. Performance assessment utilizes the VBench benchmark [2], following established protocols [3, 12] with 5-video sampling per prompt. Evaluation encompasses 16 comprehensive VBench dimensions covering aesthetic quality, temporal consistency, motion dynamics, and semantic understanding. Additional quantitative metrics include PSNR [1], SSIM [6], and LPIPS [11] to provide multi-faceted quality assessment.

Baselines. Our comparative analysis includes representative approaches from three categories: (1) sparsity-based methods (SparseVideoGen [7], STA [10]), (2) quantization-focused techniques (SageAttention [8]), and (3) joint optimization methods (SpargAttn [9]). This selection enables comprehensive evaluation of FPSAttention against both specialized single-optimization approaches and competing joint methods, providing a thorough assessment of our framework’s relative performance and efficiency gains.

A.3 Ablation Study: Challenges of Naive Quantization and Sparsity Combination

To validate our core motivation that naive combination of FP8 quantization and sparsity presents significant challenges, we conduct a comprehensive ablation study comparing three key approaches:

(1) the baseline full-precision model, (2) a training-free naive combination of quantization and sparsity, and (3) our proposed FPSAttention method with joint optimization. This comparison directly addresses the fundamental tension between quantization and sparsity mechanisms discussed in the main paper.

Table A presents a detailed comparison across all VBench metrics for the Wan 1.3B model. The training-free approach applies standard FP8 quantization and sparse attention patterns without joint optimization or denoising step-aware adaptation. As hypothesized, this naive combination leads to substantial performance degradation across nearly all evaluation metrics.

Table A: Ablation study demonstrating the challenges of naive quantization and sparsity combination. We compare baseline full-precision (Baseline), training-free naive combination (Training-Free), and our joint optimization approach (FPSAttention) on Wan 1.3B across all VBench metrics. The severe degradation in the training-free approach validates the need for holistic joint optimization.

Metric	Baseline	Training-Free	FPSAttention
Aesthetic Quality	0.6105	0.2892	0.6240
Appearance Style	0.7157	0.7874	0.7252
Background Consistency	0.9503	0.9280	0.9156
Color	0.9049	0.4836	0.8932
Dynamic Degree	0.3014	0.3750	0.4195
Human Action	0.7720	0.0200	0.7780
Imaging Quality	0.6708	0.6868	0.7103
Motion Smoothness	0.9527	0.9513	0.9413
Multiple Objects	0.6091	0.0000	0.6665
Object Class	0.7710	0.0109	0.8185
Overall Consistency	0.6453	0.1206	0.6893
Quality Score	0.8332	0.7473	0.8428
Scene	0.3030	0.0129	0.3870
Semantic Score	0.6768	0.1733	0.7088
Spatial Relationship	0.7317	0.0008	0.7659
Subject Consistency	0.9457	0.8887	0.9338
Temporal Flickering	0.9844	0.9401	0.9336
Temporal Style	0.6382	0.1239	0.6558
Total Score	0.8019	0.6325	0.8160
Performance Drop	—	-21.1%	+1.8%

Key Findings: The results clearly demonstrate the challenges inherent in naive quantization and sparsity combination:

- **Severe Quality Degradation:** The training-free approach achieves only 0.6325 total score compared to the baseline’s 0.8019, representing a substantial 21.1% performance drop.
- **Critical Failure Modes:** Several metrics show near-zero performance in the training-free approach, including Human Action (0.02), Multiple Objects (0.0), Object Class (0.011), and Spatial Relationship (0.0008), indicating complete failure in complex semantic understanding tasks.
- **Magnified Quantization Errors:** As predicted by our theoretical analysis, sparsity mechanisms amplify quantization errors in high-magnitude attention scores. This is particularly evident in metrics requiring fine-grained semantic understanding, where the interaction between quantization noise and sparse token selection leads to catastrophic information loss.
- **Joint Optimization Success:** In contrast, our FPSAttention approach not only avoids the degradation seen in naive combination but actually improves upon the baseline (0.8160 vs 0.8019, +1.8% improvement), validating the effectiveness of our denoising step-aware joint optimization strategy.

This ablation study emphasizes the necessity of our training-aware co-design scheme.

98 A.4 VBench Full Evaluation Results

99 Tables B and C present comprehensive evaluation results of our method compared to various base-
100 lines on VBench for Wan 1.3B and Wan 13B models, respectively. These results demonstrate the
101 effectiveness of our joint FP8 quantization and sparsity approach across multiple video quality
102 metrics.

103 Table B shows performance comparisons across seven methods on the Wan 1.3B model: the base-
104 line (Base), SageAttention (SageAtt) [8], SpargeAttention (SpargeAtt) [9], SparseVideoGen (Spar-
105 seVG) [7], Sliding Tile Attention (STA) [10], our quantization-only variant (Ours-Q), and our full
106 joint quantization and sparsity method (Ours-Q+S). The evaluation covers 18 comprehensive metrics
107 including aesthetic quality, motion dynamics, temporal consistency, and semantic understanding.
108 Our full method (Ours-Q+S) achieves the highest total score of 0.8160, demonstrating superior
109 performance compared to methods that apply quantization or sparsity independently.

110 Table C presents similar comparisons for the larger 13B model, where our method continues to
111 achieve competitive performance while providing substantial computational savings. The results
112 validate that our approach scales effectively to larger model sizes while maintaining video generation
113 quality across diverse evaluation criteria.

114 A.5 Clarification on vBench Evaluation Metrics

115 In this study, we observed that some of the baseline methods we reproduced (including some of our
116 own exploratory experiments prior to FPSAttention) might yield VBench scores slightly lower than
117 those reported in their respective official publications. We attribute this primarily to the following
118 factors: Randomness: The inherent stochasticity in video generation models can lead to slight
119 variations in results and vBench scores across multiple runs, even with identical settings. Prompt
120 Extension: Many prior works [4] may employ specific prompt extension strategies to enrich input
121 prompts. This can influence the content and quality scores of the generated videos. We didn’t employ
122 this optimization. Classifier-Free Guidance (CFG) Scale and Other Sampling Strategies: Different
123 CFG scale values and other sampling parameters (e.g., number of sampling steps) significantly
124 impact generation quality. While we endeavored to follow the descriptions in the respective baseline
125 papers, subtle parameter differences might still exist. It is worth noting that similar observations
126 have been made in other research. For instance, in the work on Sliding Tile Attention (STA) [10],
127 their reproduced HunyuanVideo baseline also exhibited lower VBench performance compared to
128 VBench’s official leaderboard. Despite these potential metric variations, we emphasize that all
129 methods in this study (including our FPSAttention and all compared baselines) were evaluated
130 under an identical VBench evaluation pipeline and parameter settings, ensuring a fair comparison.
131 Our primary research objective is to demonstrate the significant inference speedup achieved by
132 FPSAttention while maintaining comparable (or superior) generation quality relative to baseline
133 methods.

134 The results demonstrate that our joint FP8 quantization and sparsity approach achieves competitive
135 or superior performance compared to specialized methods focusing solely on either quantization or
136 sparsity. For the Wan 1.3B model, our method achieves the highest total score (0.8160), outperforming
137 the baseline (0.8019) while providing significant computational benefits. Similarly, for the Wan 13B
138 model, our approach performs on par with the best-performing methods while offering substantial
139 memory and compute savings through the combination of quantization and structured sparsity.

140 A.6 Training Hyperparameters

141 Table D presents the key hyperparameters used in our experiments for both Wan 1.3B and 13B model
142 training configurations. These hyperparameters were carefully selected to balance training stability,
143 convergence speed, and final model performance while accommodating the constraints imposed by
144 FP8 quantization and structured sparsity.

Table B: Performance comparison of different methods on Wan 1.3B across VBench metrics. We compare the baseline (Base), SageAttention (SageAtt), SpargeAttention (SpargeAtt), SparseVideoGen (SparseVG), Sliding Tile Attention (STA), our quantization-only variant (Ours-Q), and our full joint method (Ours-Q+S). Bold values indicate the best performance for each metric.

Metric	Base	SageAtt	SpargeAtt	SparseVG	STA	Ours-Q	Ours-Q+S
Aesthetic Quality	0.6105	0.6104	0.5668	0.563	0.5661	0.6091	0.624
Appearance Style	0.7157	0.715	0.7744	0.2253	0.7952	0.6922	0.7252
Background Consistency	0.9503	0.95	0.9123	0.9525	0.9284	0.9472	0.9156
Color	0.9049	0.8866	0.8903	0.9037	0.8974	0.9146	0.8932
Dynamic Degree	0.3014	0.307	0.3222	0.7139	0.5722	0.3222	0.4195
Human Action	0.772	0.75	0.734	0.73	0.622	0.75	0.778
Imaging Quality	0.6708	0.6699	0.6541	0.6729	0.6626	0.6798	0.7103
Motion Smoothness	0.9527	0.9527	0.9139	0.9726	0.9649	0.9496	0.9413
Multiple Objects	0.6091	0.5837	0.4715	0.471	0.4043	0.6011	0.6665
Object Class	0.7710	0.7695	0.6859	0.6935	0.5818	0.7851	0.8185
Overall Consistency	0.6453	0.6451	0.6492	0.6935	0.6236	0.6478	0.6893
Quality Score	0.8332	0.8337	0.8049	0.2336	0.7994	0.8363	0.8428
Scene	0.3030	0.3092	0.2192	0.1732	0.1995	0.3200	0.3870
Semantic Score	0.6768	0.6704	0.6412	0.6342	0.6012	0.6780	0.7088
Spatial Relationship	0.7317	0.7364	0.7217	0.6469	0.6863	0.7438	0.7659
Subject Consistency	0.9457	0.9453	0.8982	0.9292	0.8993	0.9458	0.9338
Temporal Flickering	0.9844	0.9841	0.9647	0.9883	0.9652	0.9822	0.9336
Temporal Style	0.6382	0.6385	0.6247	0.2265	0.6005	0.6475	0.6558
Total Score	0.8019	0.8011	0.7722	0.7827	0.7597	0.8046	0.8160

Table C: Performance comparison of different methods on Wan 13B across VBench metrics. We compare the baseline (Base), SageAttention (SageAtt), SpargeAttention (SpargeAtt), SparseVideoGen (SparseVG), Sliding Tile Attention (STA), and our full joint method (Ours-Q+S). Bold values indicate the best performance for each metric.

Metric	Base	SageAtt	SpargeAtt	SparseVG	STA	Ours-Q+S
Aesthetic Quality	0.6204	0.6209	0.5875	0.6246	0.6033	0.624
Appearance Style	0.2164	0.2163	0.7586	0.2306	0.2303	0.2073
Background Consistency	0.9691	0.9687	0.9355	0.9589	0.9573	0.9377
Color	0.8879	0.8825	0.8768	0.8883	0.8814	0.8932
Dynamic Degree	0.6944	0.7028	0.6028	0.6806	0.7028	0.8389
Human Action	0.796	0.8	0.78	0.816	0.778	0.816
Imaging Quality	0.6715	0.6724	0.635	0.6868	0.6577	0.7103
Motion Smoothness	0.9828	0.9828	0.9413	0.982	0.9714	0.9804
Multiple Objects	0.6627	0.6477	0.6066	0.7012	0.6576	0.666
Object Class	0.8299	0.8312	0.7896	0.8712	0.81	0.8185
Overall Consistency	0.6912	0.6912	0.6975	0.708	0.6893	0.6893
Quality Score	0.6577	0.8428	0.8134	0.7421	0.8246	0.7103
Scene	0.3669	0.3049	0.3495	0.4129	0.3387	0.3182
Semantic Score	0.7572	0.7091	0.6969	0.7421	0.7077	0.7088
Spatial Relationship	0.7364	0.7405	0.7526	0.8056	0.7405	0.7661
Subject Consistency	0.9528	0.953	0.9173	0.9489	0.953	0.9435
Temporal Flickering	0.9922	0.9922	0.969	0.9891	0.9922	0.9754
Temporal Style	0.2408	0.2408	0.6607	0.2438	0.2408	0.6558
Total Score	0.8153	0.8158	0.7901	0.8196	0.8012	0.816

145 B Limitation and Societal Impacts

146 **Limitations.** While FPSAttention demonstrates strong performance across our evaluation scenarios,
147 there are some considerations for broader adoption. Our approach works best with modern FP8-
148 capable GPUs such as NVIDIA Hopper architectures, though it can still provide benefits on older
149 hardware with reduced FP8 acceleration. The method benefits from quantization-aware training to
150 achieve optimal results, which involves a moderate increase in training time compared to post-training
151 quantization approaches. The denoising step-aware scheduling includes several hyperparameters

Table D: Comprehensive hyperparameter configuration for Wan 1.3B and 13B model training and evaluation. The table covers model architecture specifications, training parameters, diffusion scheduler settings, data configuration, and system-level precision settings used in our experiments.

Category	Parameter	Wan 1.3B	Wan 13B
Model Architecture	Model Type	WanX21FPS	WanX21FPS-13B
	Model Dimension	1536	5120
	Number of Layers	30	40
	Number of Heads	12	40
	FFN Dimension	8960	13824
	Input/Output Dimension	16	16
	Frequency Dimension	256	256
	Text Dimension	4096	4096
	Patch Size	[1, 2, 2]	[1, 2, 2]
Training	Learning Rate	5e-6	5e-6
	Weight Decay	1e-4	1e-4
	Gradient Clipping	1.0	1.0
	Warmup Steps	200	200
	EMA Decay	0.99	0.99
	Adam Epsilon	1e-15	1e-15
Diffusion Scheduler	Scheduler Type	rflow-wanx	rflow-wanx
	Number of Timesteps	1000	1000
	Sample Steps	50	50
	CFG Scale	5.0	5.0
	Sample Shift	5.0	5.0
	Transform Scale	5.0	5.0
	Sample Method	logit-normal	logit-normal
Data & Sequence	Text Length	512	512
	Max Sequence Length	75600	75600
	Sample FPS	16	16
	Video Resolution	480p	480p
	Video Duration	5s	5s
	Prompt Uncond Probability	0.1	0.1
System & Precision	Data Type	fp8	fp8
	Training Mode	FSDP	FSDP
	Gradient Checkpointing	True	True
	Quantization	True	True
	Sequence Parallel Degree	1	4

(transition points α_1 and α_2 , quantization granularities, and window sizes) that can be optimized for different model architectures and datasets. Our current evaluation focuses on the Wan2.1 architecture, and the approach shows strong promise for extension to other video diffusion transformer architectures (e.g., HunyuanVideo, CogVideoX). Additionally, while our tile-wise approach achieves good hardware utilization across tested configurations, there are opportunities for architecture-specific optimization to further improve performance on different GPU memory hierarchies.

Societal Impacts. The acceleration techniques presented in FPSAttention have both positive and potentially concerning societal implications. On the positive side, our work democratizes access to high-quality video generation by significantly reducing computational requirements, enabling broader adoption of video diffusion models for creative applications, education, and accessibility tools. The substantial speedups (up to 4.96x end-to-end acceleration) can reduce energy consumption and carbon footprint associated with video generation, contributing to more sustainable AI practices. However, the increased efficiency and accessibility of video generation technology also raise concerns about

165 potential misuse. The ability to generate high-quality videos more rapidly could facilitate the creation
166 of deepfakes, misinformation campaigns, or unauthorized synthetic content that impersonates real
167 individuals. Additionally, widespread access to efficient video generation tools may impact creative
168 industries by automating content creation processes, potentially affecting employment in video
169 production and related fields. We emphasize the importance of developing and deploying appropriate
170 safeguards, content authentication mechanisms, and ethical guidelines alongside these technological
171 advances to mitigate potential negative impacts while preserving the beneficial applications of
172 accelerated video generation.

173 **C Visualization**

174 The following qualitative comparison demonstrates that our FPSAttention method generates video
175 frames that are visually nearly identical to the baseline Wan model with size of 1.3B. This visual
176 similarity across diverse scenarios—including boats, fish, dogs, desert landscapes, couples, trains,
177 cars, cats, and robot DJs—validates that our joint FP8 quantization and sparsity optimization achieves
178 essentially lossless performance while providing substantial computational acceleration.

Table E: Qualitative comparison on the boat group. Prompt: ‘A boat sailing leisurely along the Seine River with the Eiffel Tower in background in super slow motion’ . Top: Baseline; Bottom: FPSAttention.

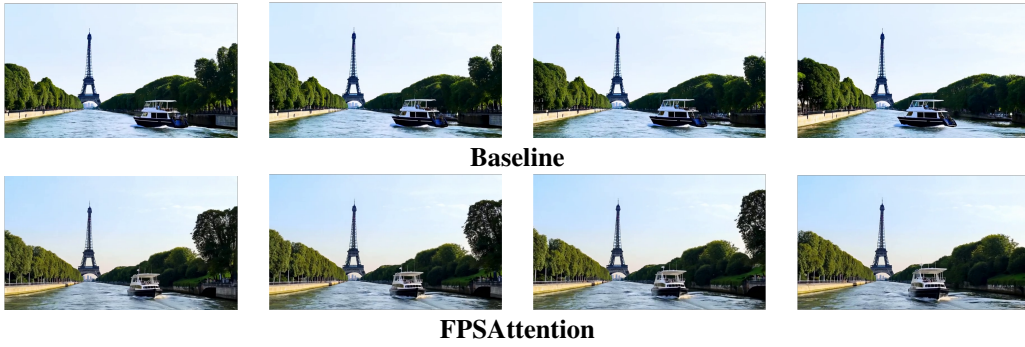


Table F: Qualitative comparison on the fish group. Prompt: ‘Golden fish swimming in the ocean’. Top: Baseline; Bottom: FPSAttention.

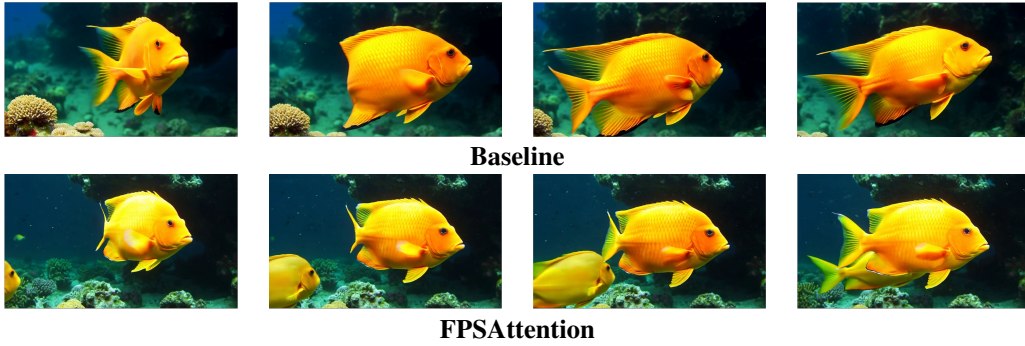


Table G: Qualitative comparison on the dog group. Prompt: ‘A dog enjoying a peaceful walk’. Top: Baseline; Bottom: FPSAttention.

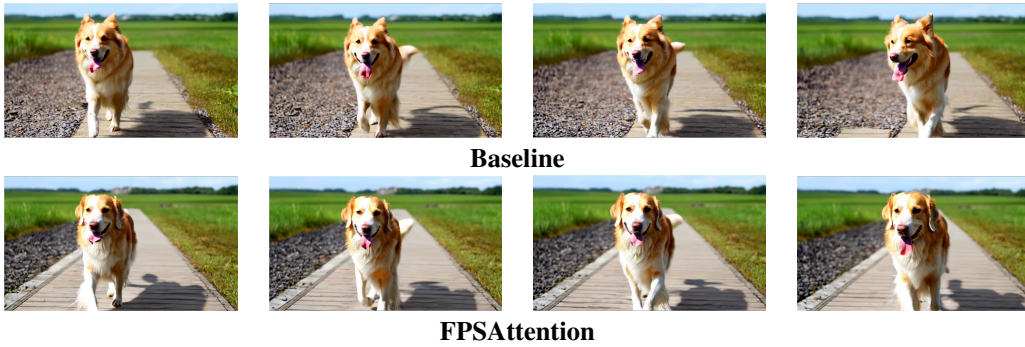


Table H: Qualitative comparison on the desert group. Prompt: ‘Static view on a desert scene with an oasis palm trees and a clear calm pool of water’. Top: Baseline; Bottom: FPSAttention.

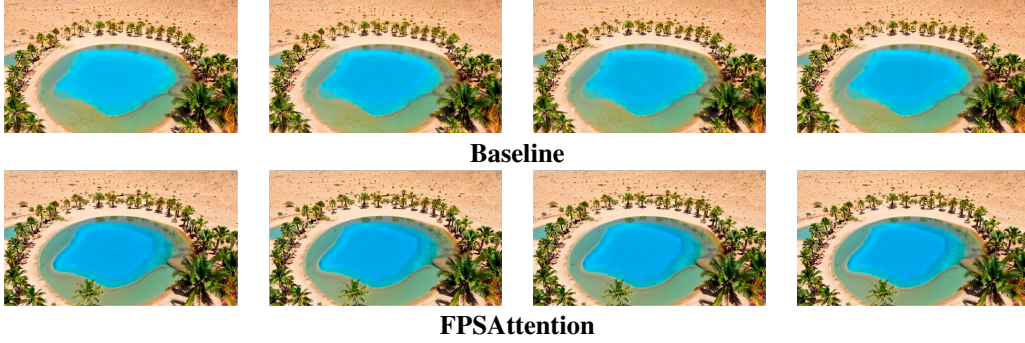


Table I: Qualitative comparison on the couple group. Prompt: ‘A couple in formal evening wear going home get caught in a heavy downpour with umbrellas’. Top: Baseline; Bottom: FPSAttention.



Table J: Qualitative comparison on the train group. Prompt: ‘A train accelerating to gain speed’. Top: Baseline; Bottom: FPSAttention.

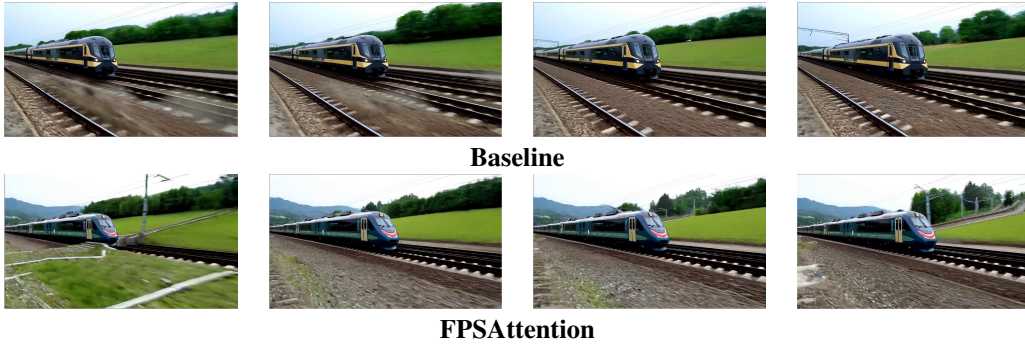


Table K: Qualitative comparison on the rock group. Prompt: ‘A tranquil tableau of at the edge of the Arabian Desert, the ancient city of Petra beckoned with its enigmatic rock-carved façades’. Top: Baseline; Bottom: FPSAttention.



Table L: Qualitative comparison on the nursery group. Prompt: ‘Nursery’. Top: Baseline; Bottom: FPSAttention.



Table M: Qualitative comparison on the snow group. Prompt: ‘Snow rocky mountains peaks canyon. snow blanketed rocky mountains surround and shadow deep canyons. The canyons twist and bend through the high elevat’. Top: Baseline; Bottom: FPSAttention.

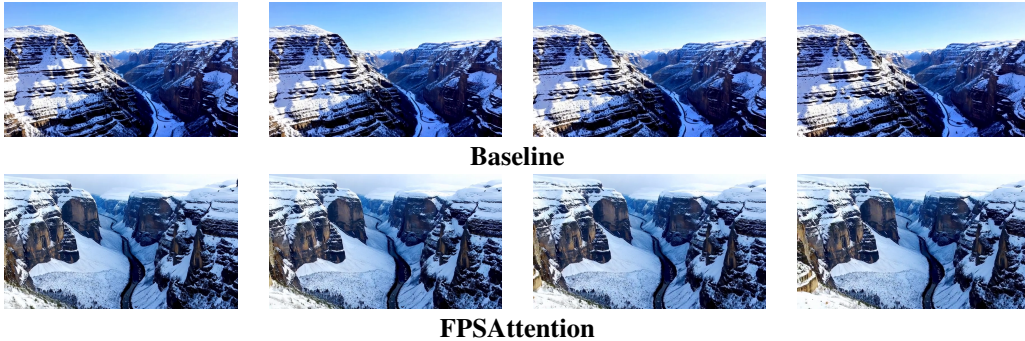


Table N: Qualitative comparison on the book group. Prompt: ‘A person is reading book’. Top: Baseline; Bottom: FPSAttention.



Table O: Qualitative comparison on the space group. Prompt: ‘An astronaut flying in space’. Top: Baseline; Bottom: FPSAttention.

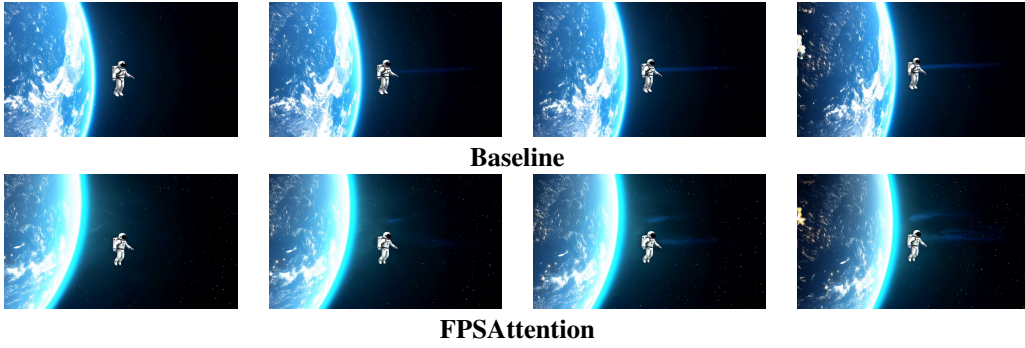
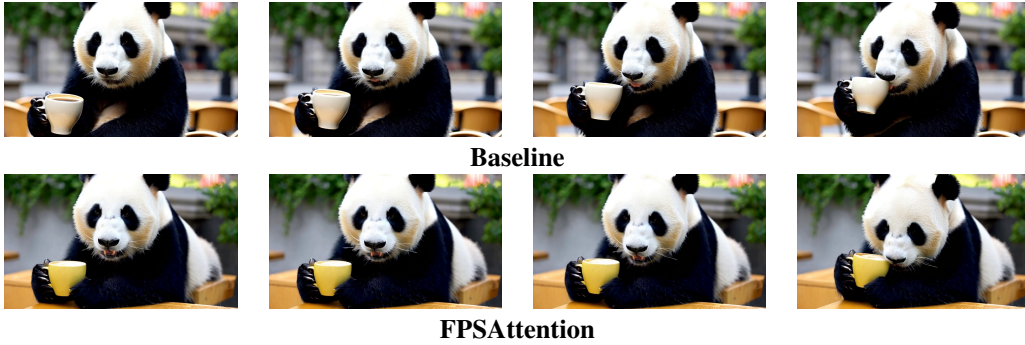


Table P: Qualitative comparison on the panda group. Prompt: ‘A panda drinking coffee in a cafe in Paris’. Top: Baseline; Bottom: FPSAttention.



References

- [1] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010.
- [2] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.
- [3] Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Kai Li, and Song Han. Distrifusion: Distributed parallel inference for high-resolution diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7183–7193, 2024.
- [4] Zefan Li et al. Cogvideox: Text-to-video diffusion models with an expert pipeline. *arXiv preprint arXiv:2408.06072*, 2024.
- [5] Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.
- [6] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [7] Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, et al. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. *arXiv preprint arXiv:2502.01776*, 2025.
- [8] Jintao Zhang, Haofeng Huang, Pengle Zhang, Jun Zhu, Jianfei Chen, et al. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. *arXiv preprint arXiv:2410.02367*, 2024.
- [9] Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattn: Accurate sparse attention accelerating any model inference. *arXiv preprint arXiv:2502.18137*, 2025.
- [10] Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhenghong Liu, and Hao Zhang. Fast video generation with sliding tile attention. *arXiv preprint arXiv:2502.04507*, 2025.
- [11] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [12] Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. *arXiv preprint arXiv:2408.12588*, 2024.