COMPUTE LESS, CAPTURE MORE: IMPROVING SPARSE ATTENTION WITH MIXED PRECISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse attention reduces computation by selecting which query–key interactions to evaluate, but typically computes every selected interaction at the same numerical precision. We find that this is inefficient: importance remains highly uneven within the selected set, while useful interactions are still omitted. We introduce **mixed-precision sparse attention**, which converts this excess precision into broader coverage. It preserves high precision for important interactions, lowers the precision of less important retained interactions, and uses the saved compute to recover interactions that the original sparse method would otherwise skip. By reusing the geometry or importance ranking of existing sparse methods, it expands coverage without a new selection mechanism or additional compute. Our analysis shows that this exchange can reduce attention-output error because broader coverage better preserves dense attention distributions and its benefit can outweigh the error introduced by lower precision. We further develop a fused kernel that executes BF16, FP8, and FP4 interactions within a single attention operation with little overhead. Across video generation and long-context language understanding, mixed-precision sparse attention consistently improves the quality–compute frontier of existing sparse attention methods.

1 INTRODUCTION

Sparse attention reduces attention cost by deciding which query–key interactions are worth computing. Recent methods use increasingly sophisticated geometry, clustering, or input-dependent importance estimates to make this decision in long-context language models and video generation (Li et al., 2025; Yang et al., 2025; Lai et al., 2025; Xu et al., 2025). Yet after carefully deciding where computation should be spent, most methods make a much coarser arithmetic decision: all selected interactions are evaluated at the same numerical precision. In other words, sparse attention already solves a resource-allocation problem, but only along one dimension—**where to compute**. It leaves a second dimension largely unused—**how precisely to compute**.

We show that this missing dimension presents much optimization room. Within the query–key interactions selected by existing sparse attention algorithms, importance is uneven: across five video and language models, the highest-ranked 10% of selected interactions account for approximately 50–78% of the attention mass within the selected set. In other words, sparsification does not eliminate importance imbalance; interactions that contribute very differently to the attention output are still computed at the same precision. This motivates a simple question: **Can sparse attention spend less precision on lower-importance interactions and use the saved compute to recover useful interactions it would otherwise skip?**

We introduce **mixed-precision sparse attention**, which jointly reallocates computation across coverage and numerical precision. Rather than designing a new sparse selection method, we reuse the geometry or importance ranking already produced by established sparse attention methods. We preserve higher precision for the most important interactions, compute less important selected interactions at lower precision, and spend the saved compute on additional interactions that were originally skipped. The resulting policy therefore assigns interactions to BF16, FP8, FP4, or skip under the original sparse method's compute budget. Improvements in sparse selection and mixed-precision allocation are consequently complementary rather than competing directions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

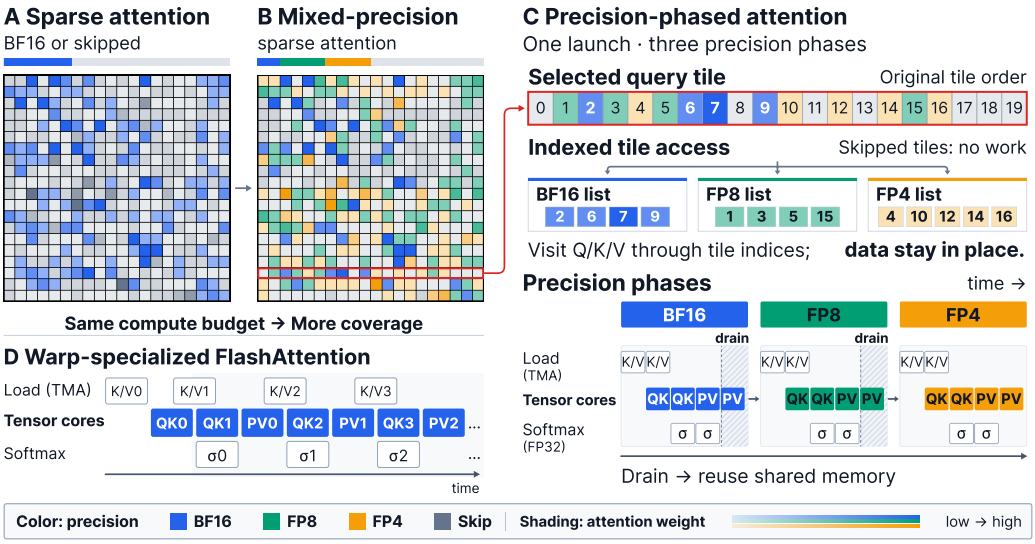


Figure 1: **Sparse and mixed-precision attention at equal wall-clock time. (A–B)** Sparse attention computes selected query–key pairs in BF16 and skips the rest; mixed precision additionally uses FP8 and FP4 to cover more pairs under the same compute budget. Color denotes precision used; **darker shading** denotes larger attention weight, and gray denotes skipped interactions. **(C)** A fused precision-phased kernel processes key/value tiles grouped by precision in one launch, carrying a shared online-softmax and output state across phases in registers. Tiles are accessed through indices without rearranging the $Q/K/V$ tensors, and shared-memory buffers are reused between phases. **(D)** Standard warp-specialized FlashAttention pipeline.

The key question is whether the information recovered through this broader coverage is worth the numerical error introduced by lower precision. Using an incremental attention-error update, we find lower-error mixed-precision allocations in nearly all tested cases under the corresponding sparse-attention compute budget. We further study why this opportunity exists at two levels. First, mixed precision more closely preserves the dense post-softmax attention distribution by recovering interactions that sparse attention omits. Second, the numerical disturbance introduced by low precision is typically smaller than the useful value contribution recovered by computing those interactions.

To support these allocations in practice, we further develop a fused precision-phased attention kernel that executes multiple numerical precision tiers within a single attention operation with little additional overhead. This shows that fine-grained precision allocation can be realized efficiently.

Across video generation and long-context language understanding, mixed precision improves the quality–compute tradeoff in two complementary ways: higher quality at the same attention compute, and lower compute at the same generation quality. On Wan2.1-14B and HunyuanVideo-13B, mixed Radial and SVG2 improve PSNR by 1.90–6.39dB at matched attention compute. On 64K RULER, mixed precision reduces attention compute by 31.3–39.1% at matched quality targets for XAttention,FlexPrefill and SeerAttention. Together, these results show that precision provides a new resource-allocation dimension for sparse attention: reallocating compute between precision and coverage improves the quality–compute frontier of existing methods.

Our contributions are:

- **A new allocation dimension for sparse attention.** We identify importance imbalance within the interactions selected by existing sparse attention algorithms and introduce mixed-precision sparse attention, which jointly allocates both coverage and precision.
- **Analysis of why mixed precision helps.** We show that lower-error mixed precision allocations exist broadly under matched compute budgets and study their mechanism.
- **Practical mixed-precision kernel.** We develop a fused attention kernel supporting multiple precision tiers with little overhead and demonstrate consistent improvements in the quality–compute frontier across video generation and long-context language tasks.

2 Related Work

Sparse attention. Sparse attention reduces computation by selecting a subset of query–key interactions, using either predefined structure or input-dependent importance score. *Geometry-drive methods* exploit locality or spatiotemporal patterns (Xiao et al., 2024; Zhang et al., 2025d; Li et al., 2025), while *Input-adaptive methods* construct sparse patterns from the current activations using learned scores, clustering, or dynamic indices (Jiang et al., 2024; Lai et al., 2025; Xu et al., 2025; Tang et al., 2024; Yang et al., 2025). Rather than introducing a new selection mechanism, we build on these well-established methods and reuse their existing geometry or importance rankings to additionally determine *how precisely* each interaction is computed.

Low-precision attention. Low-precision attention reduces arithmetic cost by replacing BF16 computation with lower-precision formats: FlashAttention-3 introduces FP8 attention kernels, while SageAttention and SageAttention3 develop efficient INT8/FP8 and FP4 attention implementations (Shah et al., 2024; Zhang et al., 2025b;a). SpargeAttention and FPSAttention further combine quantization with sparse computation (Zhang et al., 2025c; Liu et al., 2025). Other work varies precision within a workload for different purposes: MiKV assigns different bit widths to cached tokens to reduce KV-cache memory (Yang et al., 2024), while ThriftAttention keeps dense attention coverage but promotes selected blocks from FP4 to BF16 (Sharratt, 2026). In contrast, we use multiple precision levels within existing sparse attention to jointly allocate precision and coverage, recovering more interactions under the same compute budget.

3 Motivation and Overview

The usefulness of mixed-precision allocation depends on whether sparse attention has meaningful variation of importance among the interactions it selects. We find that such variation is common. Figure 2 ranks the query–key interactions selected by each sparse method (Yang et al., 2025; Li et al., 2025; Lai et al., 2025) from low to high by their block-average post-softmax attention weight and plots their cumulative attention mass. If the selected interactions contributed uniformly, the curves would follow the diagonal. Instead, the curves rise sharply only near the highest-ranked interactions: the top 10% alone account for 49.6%–78.2% of the attention mass within the selected set. This concentration appears consistently across both video and language models and across different sparse attention algorithms.

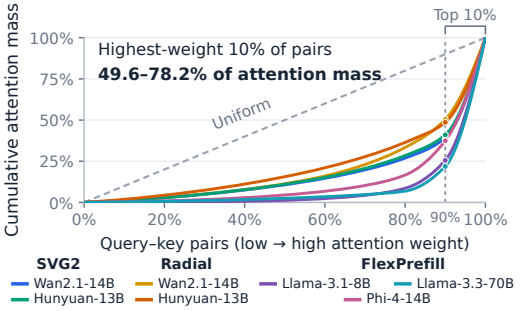


Figure 2: **Attention mass is concentrated within sparse selections.** Retained query–key pairs are ordered by increasing attention weight. The diagonal represents uniform weights.

Figure 1A illustrates this imbalance. Color intensity denotes attention weight, with darker cells indicating more important query–key interactions. Although all selected interactions are computed at the same precision, their importance varies substantially. Moreover, some high-weight interactions still lie outside the region selected by the sparse method and are skipped entirely.

Figure 1B illustrates our solution. We keep higher precision for the most important interactions, reduce precision on less important selected interactions, and use the saved compute to recover additional interactions that the original sparse policy would skip. This jointly reallocates precision and coverage under the same compute budget, allowing mixed precision to capture a broader set of important query–key interactions. Hue denotes the assigned precision, while shading indicates attention weight. Compared with the original sparse allocation in Figure 1A, mixed precision better aligns computation with the underlying importance distribution.

4 Mixed-Precision Allocation from Existing Sparse Attention

We construct mixed-precision allocations by reusing each sparse attention's existing importance signal. Rather than introducing a new importance estimator, we use these signals to assign each

interaction to BF16, FP8, FP4, or skip, while matching the original method's compute budget for a fair comparison with existing sparse attention algorithms (Figure 3).

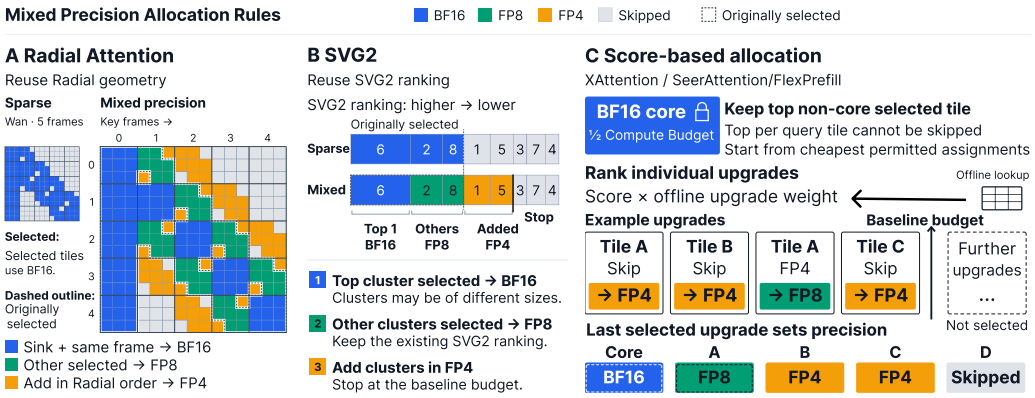


Figure 3: **Mixed-precision allocation from existing sparse attention.** We reuse each method's original importance signal to assign precision: **(A)** Radial's geometry-based ordering, **(B)** SVG2's cluster ranking, and **(C)** FlexPrefill/XAttention/SeerAttention's tile-importance scores.

Radial Attention (Figure 3A). We preserve Radial's geometric ordering (Li et al., 2025). The first-frame attention sink and same-frame interactions are computed with BF16 precision. Other interactions selected by Radial Attention are computed in FP8. The computation saved by using FP8 is then used to include previously skipped interactions in FP4, following the same Radial ordering from more important to less important interactions. We continue adding FP4 interactions until the available compute budget is exhausted.

SVG2 (Figure 3B). We reuse SVG2's clustering and its ranking of query-key clusters (Yang et al., 2025). Among the clusters selected by SVG2, the highest-ranked selected cluster is kept in BF16, while the remaining clusters are computed in FP8. We then spend the saved computation on previously skipped key clusters in FP4, following SVG2's original ranking from highest to lowest. In the evaluation, for a fair comparison, we stop when the available compute budget is exhausted.

XAttention / SeerAttention / FlexPrefill (Figure 3C). We reuse each method's selected tiles and importance scores (Xu et al., 2025; Gao et al., 2025). For each head, we reserve half of the arithmetic budget for a BF16 core containing mandatory tiles (sink and causal-diagonal tiles). We then protect the best remaining key tile for each query tile from being skipped, ensuring that important row-specific interactions are retained. The remaining tiles are allowed to vary in precision, enabling the allocation to balance refinement of selected interactions with additional coverage.

For the purpose of equal-compute evaluation, we use fixed offline coefficients to allocate the remaining budget. The cost c_p of precision p is the per-tile kernel cost at that precision, normalized by the kernel time of processing an equal-sized tile in BF16 ($c_{\text{BF16}} = 1$, $c_{\text{skip}} = 0$); the error penalty e_p is derived from relative L_2 attention-output error. For an upgrade with selector score s, error reduction Δe, and added cost Δc, we rank it by $s \cdot \Delta e / \Delta c$. Starting from each tile's cheapest permitted assignment, we apply upgrades in descending priority until the sparse method's compute budget is exhausted. Same tiles may therefore progress from FP4 to FP8 to BF16.

5 Analysis of Mixed-Precision Sparse Attention

The previous section provides practical rules for constructing mixed-precision allocations. But does this additional flexibility actually produce more accurate attention outputs under the same compute budget, and if so, why? Let B denote the original binary sparse allocation, where each interaction is either computed in BF16 or skipped, and let $C(B)$ denote its compute cost. We ask whether there exists a mixed-precision allocation X such that $C(X) \leq C(B)$ while reducing its attention-output error relative to dense attention.

Attention and output error. For one query, dense attention is $O = \sum_j \pi_j v_j$, where $\pi_j = e^{s_j} / \sum_\ell e^{s_\ell}$ is the post-softmax attention weight. A mixed-precision allocation X assigns each tile i to one of BF16, FP8, FP4, skip; we write $X(i)$ for the precision assigned to tile i. At precision p, define the tile's unnormalized attention mass and weighted-value contribution as $w_i^{(p)} = \sum_{j \in i} e^{s_j^{(p)}}$ and $a_i^{(p)} = \sum_{j \in i} e^{s_j^{(p)}} v_j^{(p)}$. The output produced by mixed precision allocation X is normalized by its total attention mass $W_X = \sum_i w_i^{(X(i))}$, and its error relative to dense attention can be written as

$$\widetilde{O}_X - O = \frac{N_X}{W_X}, \qquad N_X = \sum_i \left(a_i^{(X(i))} - w_i^{(X(i))} O \right), \qquad W_X = \sum_i w_i^{(X(i))}. \tag{1}$$

Here, W_X is the softmax normalizer induced by X, while N_X measures the total weighted error before normalization. Dividing N_X by W_X gives the final attention-output error.

Lower-error allocations. Under the same compute budget, we ask whether reallocating precision and coverage can reduce attention-output error. Evaluating each changed tile independently is insufficient because softmax couples all interactions through a shared normalization, so one edit changes the entire softmax distribution. Meanwhile, replaying attention for every candidate repeatedly recomputes unchanged tiles and becomes impractical when many reallocations must be evaluated.

We exploit the additivity of unnormalized attention contributions to maintain a compact state for the current allocation. The pair (N_B, W_B) summarizes its weighted error and normalization, so a candidate only needs to update the tiles it changes. Let B be the current binary sparse allocation and M a candidate allocation. If the edits change the weighted-value sum by ΔA and the attention-mass sum by ΔW, then

$$\frac{\|N_B + \Delta A - \Delta W\, O\|_2}{W_B + \Delta W} < \frac{\|N_B\|_2}{W_B} \tag{2}$$

exactly determines whether M has lower attention-output error than B. This turns the global softmax coupling into an exact update over only the changed tiles' contributions, avoiding recomputation of the unchanged attention state. We use this test to search for lower-error allocations within the original sparse method's compute budget; Figure 4(a) reports how often such an allocation is found.

(a) Lower-error allocations (%)

Video	Radial	SVG2
Wan2.1-14B	99.9	99.6
Hunyuan-13B	99.8	99.6

Language	FlexPrefill	XAttention	Seer
Llama-3.1-8B	100.0	99.6	99.8
Llama-3.3-70B	98.5	99.8	—
Phi-4-14B	100.0	100.0	—

—: SeerAttention checkpoint unavailable

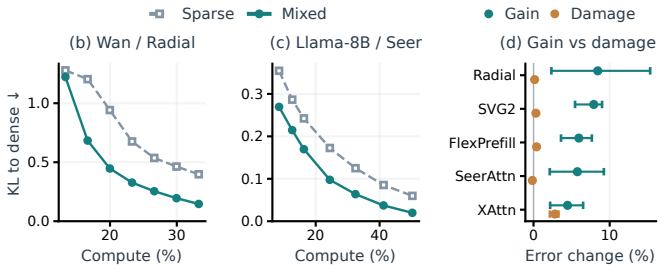


Figure 4: **Lower-error allocations and their mechanisms.** **(a)** Fraction of 1,000 sampled query tiles per model–method pair with a lower-error mixed allocation at the same compute budget. **(b–c)** KL divergence of sparse and mixed-precision attention weights from dense attention across compute budgets; lower is better. **(d)** Attention-output error reduction from added coverage (*gain*) versus error introduced by lower precision (*damage*).

Attention-distribution preservation. Finding lower-error allocations shows that mixed precision creates better operating points, but not why. We first ask whether it better preserves how dense attention distributes weight across interactions. Let $p_j = e^{s_j} / \sum_\ell e^{s_\ell}$ denote the dense post-softmax distribution. For a mixed-precision allocation X computing interactions S_X, its distribution is

$$q_j^X = \frac{\mathbf{1}[j \in S_X]\, e^{s_j^{(X(j))}}}{\sum_{\ell \in S_X} e^{s_\ell^{(X(\ell))}}}, \qquad D_{\mathrm{KL}}(q^X \| p) = \sum_j q_j^X \log \frac{q_j^X}{p_j}.$$

Thus, lower KL means that the mixed allocation preserves the dense attention-weight distribution more closely, accounting for both skipped interactions and score perturbations from lower precision.

Figures 4(b–c) show that mixed precision reduces mean KL by 4.4–63.2% for Wan/Radial and 24.0–66.9% for Llama-3.1-8B/SeerAttention (Gao et al., 2025).

Coverage gain and quantization damage. We next separate the benefit of expanding coverage from the error introduced by lower precision. Let O_S be the original sparse output, O_Q use the same sparse interactions but apply the mixed-precision allocation, and O_M be the final mixed-precision output. With dense attention O as the reference, define $E(Y) = \|Y - O\|_F$. We measure

$$\text{Damage} = \frac{E(O_Q) - E(O_S)}{\|O_S\|_F}, \qquad \text{Gain} = \frac{E(O_Q) - E(O_M)}{\|O_S\|_F}.$$

Thus, *damage* measures the error added by lower precision before changing coverage, while *gain* measures the error reduction from the mixed allocation's change in coverage. Figure 4(d) shows that the gain exceeds the quantization damage on average for all five evaluated sparse attention methods.

6 Fused Mixed-Precision Execution

Section 4's mixed-precision allocation produces a tile-level precision map M with the same grid as the tiled attention matrix. Each entry M_{ij} specifies whether query–key tile (i, j) is skipped or computed in BF16, FP8, or FP4; the assigned precision is used for both score computation and weighted-value accumulation. We extend the FlashAttention interface (Dao, 2024; Shah et al., 2024) to accept this map, $(Q, K, V, M) \mapsto \widetilde{O}_M$, where $M_{ij} \in \{\text{skip}, \text{FP4}, \text{FP8}, \text{BF16}\}$.

Figure 1C illustrates our *precision-phased attention* execution. Processing tiles in their original order would repeatedly switch precision, requiring either all precision-specific operands to coexist in shared memory or frequent reloads from HBM. Instead, we convert M into per-precision index lists and process the selected tiles in BF16, FP8, and FP4 phases. Each phase accesses its tiles through these indices without physically rearranging the Q, K, or V tensors; after a phase drains, its shared-memory buffers can be reused by the next precision's phase.

The phases share the same online-softmax state (Milakov & Gimelshein, 2018; Dao, 2024). We keep the running maximum, normalization sum, and output accumulator in registers across phases, avoiding repeated writes to and reloads from HBM.

7 Evaluation

Evaluation setup. We evaluate video-generation quality and long-context language understanding, then examine attention-output error and execution efficiency. For video, we use Wan2.1-T2V-14B (Wan Team, 2025) and HunyuanVideo-13B (Kong et al., 2024) with prompts from the Penguin Video Benchmark (Tencent Hunyuan Team, 2025), also used by SVG2 (Yang et al., 2025). We measure fidelity to dense generation using PSNR, SSIM, LPIPS and relative L_2 error, and reference-free quality using VisionReward (Xu et al., 2026) and VBench (Huang et al., 2024). For language, we evaluate Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Llama-3.3-70B-Instruct (Meta AI, 2024) and Phi-4-14B (Abdin et al., 2024) on LongBench v1 (Bai et al., 2024) and RULER (Hsieh et al., 2024), focusing on long-context prefill, where attention cost grows quadratically with sequence length (Lai et al., 2025; Xu et al., 2025).

Experiments are performed on an NVIDIA GeForce RTX 5090 (Blackwell). For both video and language, sparse and mixed attention are compared under matched compute budgets, defined by equal attention-kernel wall-clock time. We use kernel time rather than end-to-end latency because total runtime also includes method-specific costs such as importance estimation, ranking, and sparse-pattern construction, which vary across sparse attention methods and implementations, and hence is not directly comparable. Quality–compute frontier plots report attention-kernel wall-clock time relative to dense BF16 attention, with dense BF16 normalized to 100%.

7.1 Video generation

We evaluate mixed-precision Radial Attention (Li et al., 2025) and SVG2 (Yang et al., 2025), reusing Radial's geometric ordering and SVG2's cluster ranking as described in Section 4. We first examine the quality–compute frontier, followed by video-quality results at matched compute budgets.

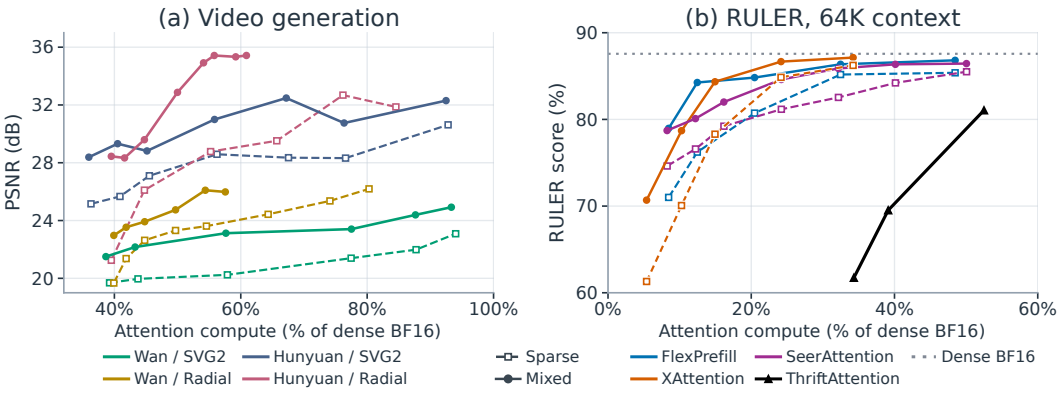


Figure 5: **Quality–compute frontiers. (a)** Video fidelity to dense generation measured by PSNR. **(b)** RULER score at 64K context on Llama-3.1-8B. Curves farther toward the upper left indicate better quality at lower costs. Solid and dashed curves denote mixed-precision and sparse attention.

Quality–compute frontier. Figure 5(a) plots PSNR to dense generation against attention-kernel wall-clock time, normalized to dense BF16 attention kernel time. Across the tested budgets, the mixed-precision curves consistently lie above their sparse counterparts, producing videos that more closely match dense generation at the same compute budget. Conversely, at matched PSNR targets, mixed precision reduces attention-kernel wall-clock time by 22.8–39.8%. Mixed-precision attention improves the video quality–compute frontier of the original sparse attention methods.

Video quality at matched compute. Having established the quality–compute frontier, we next fix representative operating points and compare video quality under matched attention-kernel wall-clock budgets. Table 1 reports results for Radial at $\beta = 0.75$ and SVG2 at top-$p = 0.70$, using each method's native sparsity-control parameter. Here, β controls Radial's geometric allocation, while top-p sets SVG2's cumulative estimated attention-mass threshold; within each comparison, sparse and mixed attention receive the same compute budget. We define *coverage* as the fraction of query–key interactions that are evaluated rather than skipped.

Table 1: **Video quality at matched attention kernel wall-clock time.**

Setting			Attention		Fidelity to dense				Reference-free				
Model	Selector	Method	Coverage	Cost	PSNR↑	SSIM↑	LPIPS↓	rel-L2↓	VR↑	Subj.↑	Bkg.↑	Flick.↑	Imag.↑
Wan2.1 14B	**Radial** $\beta = .75$	Dense	1.000	100.0%		*reference*			.089	.936	.957	.985	.608
		Sparse	.196	39.9%	22.51	.714	.238	.253	.083	.921	.948	.984	**.608**
		Mixed	.301	39.9%	**26.73**	**.845**	**.133**	**.157**	**.089**	**.932**	**.954**	**.985**	.599
	SVG2 $p = .70$	Dense	1.000	100.0%		*reference*			.092	.938	.955	.985	.601
		Sparse	.186	25.9%	21.22	.709	.276	.288	.074	.935	.953	.985	.577
		Mixed	.342	25.6%	**23.12**	**.769**	**.206**	**.238**	**.085**	**.936**	**.956**	.985	**.597**
Hunyuan 13B	**Radial** $\beta = .75$	Dense	1.000	100.0%		*reference*			.094	.937	.956	.990	.639
		Sparse	.206	39.7%	22.85	.733	.251	.200	.089	.919	.950	.990	.635
		Mixed	.395	39.7%	**29.23**	**.900**	**.087**	**.100**	.089	**.936**	**.956**	**.991**	.635
	SVG2 $p = .70$	Dense	1.000	100.0%		*reference*			.095	.938	.956	.990	.641
		Sparse	.167	26.3%	25.79	.847	.136	.145	.083	.936	.952	.991	.635
		Mixed	.319	26.2%	**29.32**	**.916**	**.074**	**.103**	**.092**	**.937**	**.954**	.991	**.639**

Mixed precision expands coverage by 1.53–1.91× and improves mean PSNR by 1.90–6.39 dB, with gains in SSIM, LPIPS and relative L_2 in all settings. VBench subject and background consistency improve in all four settings, and VisionReward improves in three. These results show that the improved attention allocation translates into higher-quality video generation at matched compute.

7.2 Long-context language understanding

We next evaluate whether mixed precision improves the quality–compute frontier for long-context language tasks. We apply mixed precision to FlexPrefill, XAttention, and SeerAttention (Lai et al.,

2025; Xu et al., 2025; Gao et al., 2025) while retaining each method's original importance scores and structural rules, and evaluate on LongBench and RULER.

Quality–compute frontier. Figure 5(b) plots RULER score at 64K context on Llama-3.1-8B against different compute budgets. We compare mixed and sparse variants of FlexPrefill, XAttention, and SeerAttention, and additionally include ThriftAttention as a low-precision attention baseline. Across the tested operating points, mixed precision consistently improves the quality–compute frontier of the corresponding sparse methods. At matched quality targets, mixed precision reduces the required compute budget by 31.3–39.1%, showing that the same long-context quality can be achieved with less attention computation with mixed-precision sparse attention.

ThriftAttention has a relatively high compute floor because every attention tile is computed at least in FP4. It then uses its own block-level promotion rule rather than the importance signals from established sparse attention methods, which can make its precision allocation less accurate.

Effect of context length. Figure 6 evaluates XAttention and its mixed-precision variant on RULER with Llama-3.1-8B as the context length increases from 16K to 128K. Mixed precision improves RULER performance across most settings, showing that the benefit of mixed-precision allocation persists at different context lengths.

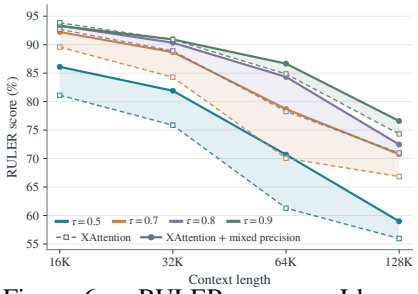


Figure 6: RULER score on Llama-3.1-8B at different context lengths using XAttention at different threshold τ.

Quality across tasks and models. We next evaluate whether the gains hold across models and long-context tasks. Table 2 compares sparse and mixed attention under matched compute budgets on different models. For each sparse method, we use the same operating point for its sparse and mixed variants. For FlexPrefill, $\gamma = 0.85$ is its attention-coverage target, which determines how many blocks are retained by its native structured sparse pattern. For XAttention, $\tau = 0.5$ is applied directly to its estimated block scores: blocks are ranked and retained until their cumulative score reaches 50% of the total. For SeerAttention, density $= 0.16$ specifies the fraction of blocks retained by its learned gate; we evaluate it on Llama-3.1-8B, where the corresponding learned gate is available. The table reports LongBench's six task categories and RULER's four task families.

Table 2: **Long-context benchmark results across models and sparse attention algorithms.**

Model / selector / method			LongBench						RULER			
Model	Selector	Method	Single QA	Multi QA	Summ.	Few-shot	Synth.	Code	NIAH	Trace	Words	QA
Llama 3.1-8B		Dense	43.00	46.55	28.88	69.18	54.80	61.81	99.51	98.75	71.77	72.27
	FlexPrefill	Sparse	41.64	40.48	28.50	**67.70**	43.50	**64.53**	96.83	96.09	62.36	70.31
	$\gamma = .85$	**Mixed**	**42.94**	**44.01**	**28.67**	67.58	**48.00**	61.91	**98.83**	**98.13**	**71.91**	**70.70**
	XAttention	Sparse	34.20	35.61	28.56	66.13	37.25	**63.62**	83.13	55.78	70.22	57.03
	$\tau = .5$	**Mixed**	**38.88**	**40.80**	**28.57**	**68.01**	**42.55**	63.49	**86.87**	**93.59**	**74.90**	**60.16**
	SeerAttention	Sparse	42.23	41.82	26.98	**67.80**	50.98	48.13	94.80	94.38	**68.21**	66.89
	Density = 0.16	**Mixed**	**42.29**	**42.80**	**27.49**	67.22	**54.22**	**54.81**	**97.53**	**97.03**	67.71	**67.58**
Llama 3.3-70B		Dense	46.35	57.69	28.34	71.17	55.41	52.77	99.75	100.00	95.68	77.08
	FlexPrefill	Sparse	43.08	52.98	28.36	71.65	40.07	45.93	99.44	99.67	79.28	69.17
	$\gamma = .85$	**Mixed**	**46.53**	**55.15**	**28.59**	**71.97**	**49.65**	**49.02**	**99.90**	**100.00**	**89.07**	**70.83**
	XAttention	Sparse	36.36	45.41	**28.89**	72.94	35.83	42.65	74.80	31.25	54.17	**60.42**
	$\tau = .5$	**Mixed**	**43.00**	**51.74**	28.34	**74.48**	**52.86**	**48.08**	**79.88**	**99.38**	**76.67**	57.29
Phi-4 14B		Dense	36.46	28.26	27.07	69.94	35.33	68.51	94.24	100.00	94.53	76.95
	FlexPrefill	Sparse	34.47	26.35	26.97	69.44	27.58	68.10	88.79	99.53	90.66	**78.91**
	$\gamma = .85$	**Mixed**	**36.34**	**27.37**	**27.03**	**70.74**	**32.38**	**68.31**	**93.90**	**100.00**	**93.78**	78.13
	XAttention	Sparse	30.15	23.97	26.70	65.20	17.24	65.17	79.89	64.71	91.98	69.85
	$\tau = .5$	**Mixed**	**33.09**	**24.77**	**27.23**	**67.56**	**34.53**	**68.45**	**84.32**	**99.71**	**93.91**	**72.79**

Mixed precision improves most categories across models and sparse attention methods, while remaining close to the sparse baseline in the few cases where it does not improve. The gains span different types of long-context tasks rather than being concentrated in a particular benchmark or

task category. Together with the quality–compute frontier, these results show that mixed precision provides consistent benefits across models, sparse attention methods, and different context lengths.

7.3 Ablations: attention error and kernel efficiency

We examine whether the practical mixed-precision allocations from Section 4 improve attention-output accuracy, and whether its fused execution preserves the efficiency of single-precision kernels.

(a) Attention-error reduction (%)

Video	Radial	SVG2
Wan2.1-14B	46.3	68.0
Hunyuan-13B	39.4	48.5

Language	FlexPrefill	XAttn	Seer
Llama-3.1-8B	42.2	24.5	35.2
Llama-3.3-70B	45.4	15.9	—
Phi-4-14B	47.5	26.4	—

—: Seer checkpoint unavailable

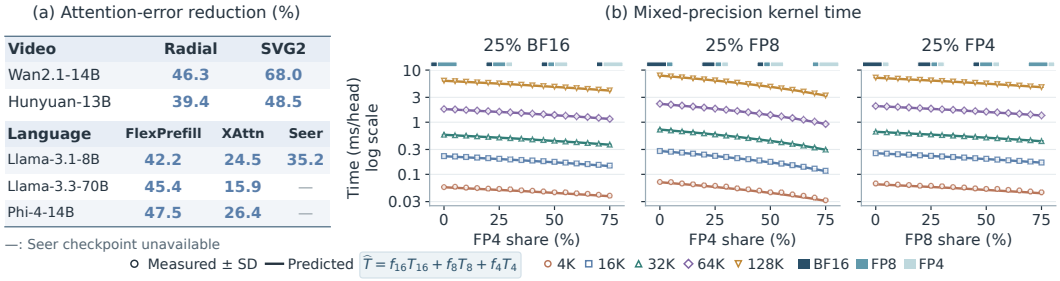


Figure 7: **Attention accuracy and execution efficiency. (a)** Attention-output error reduction from sparse to mixed precision (%); larger values indicate greater improvement. **(b)** Mixed-precision kernel time on RTX 5090 across precision mixtures and context lengths. Markers show measurements, solid lines show additive predictions from single-precision reference timings.

Attention-output error. We directly measure how closely sparse and mixed-precision attention approximate dense attention output on 1,000 captured query-tile instances per model–method pair. Figure 7(a) reports the reduction in attention-output error for Radial and SVG2 on video models, and FlexPrefill, XAttention, and SeerAttention on language models. Mixed precision reduces mean attention-output error by 15.9–68.0% across all 11 evaluated combinations. Hence, the downstream quality improvements are accompanied by consistently more accurate attention outputs.

Mixed-precision kernel efficiency. We test whether the fused kernel preserves single-precision execution efficiency when tiles assigned to each precision are noncontiguous. For 4K, 16K, and 32K contexts, we keep the same set of query–key tiles and vary only the fractions executed in BF16, FP8, and FP4. We predict runtime as $\widehat{T} = f_{16}T_{16} + f_8T_8 + f_4T_4$, where f_p is precision p's share of computed tiles and T_p is the single-precision reference time for the same amount of attention work. These references use FlashAttention for BF16 and FP8, and SageAttention for FP4.

Figure 7(b) compares this prediction with measured runtime under scattered tile locations and shuffled precision assignments. The median and mean absolute relative deviations are 1.01% and 1.92%, respectively. The close agreement shows that execution time approximately follows the precision mixture, without requiring tiles of the same precision to be contiguous in memory.

8 Conclusion

We introduced mixed-precision sparse attention, which extends sparse attention from deciding only *where* to compute to jointly allocating *coverage* and *precision*. Rather than introducing a new selection mechanism, we reuse the geometry or importance rankings of existing sparse methods: important interactions retain high precision, less important selected interactions use lower precision, and the saved compute recovers additional interactions that would otherwise be skipped.

Our analysis shows that this broader coverage better preserves dense attention and reduces attention-output error under the same compute budget. We further realize these allocations with a fused precision-phased attention kernel and demonstrate improved quality–compute tradeoffs across video generation and long-context language understanding. Together, these results show that sparse attention can use its compute budget more effectively by jointly reallocating coverage and precision.

AI USE STATEMENT

We used generative AI tools to assist with manuscript polishing, code development, mathematical derivations, and discussion of experimental design. We also used these tools to draft portions of the manuscript from author-provided outlines and technical content; all such drafts were subsequently reviewed, revised, and integrated by the authors. The research questions, central ideas, methodology, and final technical claims originated with and were determined by the authors. The authors take full responsibility for the accuracy, originality, and integrity of the final manuscript, including its text, code, analyses, and conclusions.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024. URL https://arxiv.org/abs/2412.08905.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/98ed250b203d1ac6b24bbcf263e3d4a7-Abstract-Conference.html. arXiv:2307.08691.

Yizhao Gao, Zhichen Zeng, DaYou Du, Shijie Cao, Peiyuan Zhou, Jiaxing Qi, Junjie Lai, Hayden So, Ting Cao, Fan Yang, and Mao Yang. SeerAttention: Self-distilled attention gating for efficient long-context prefilling. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/50e9dbc4ab68d94f15261ddc26c8ca2b-Abstract-Conference.html.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_VBench_Comprehensive_Benchmark_Suite_for_Video_Generative_Models_CVPR_2024_paper.html.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5dfbe6f5671e82c76841ba687a8a9ecb-Abstract-Conference.html.

Weijie Kong, Qi Tian, Zijian Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. URL https://arxiv.org/abs/2412.03603.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. FlexPrefill: A context-aware sparse attention mechanism for efficient long-sequence inference. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/03645743ea35690f30d795d6bac149a5-Abstract-Conference.html.

Xingyang Li, Muyang Li, Tianle Cai, Haocheng Xi, Shuo Yang, Yujun Lin, Lvmin Zhang, Songlin Yang, Jinbo Hu, Kelly Peng, Maneesh Agrawala, Ion Stoica, Kurt Keutzer, and Song Han. Radial attention: $o(n \log n)$ sparse attention with energy decay for long video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/18a367688479c1b08b001584218a4443-Abstract-Conference.html. arXiv:2506.19852.

Akide Liu, Zeyu Zhang, Zhexin Li, Xuehai Bai, Yuanjie Xing, Yizeng Han, Jiasheng Tang, Jichao Wu, Mingyang Yang, Weihua Chen, Jiahao He, Yuanyu He, Fan Wang, Gholamreza Haffari, and Bohan Zhuang. FPSAttention: Training-aware FP8 and sparsity co-design for fast video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ca5cfc233a6ae15e62589650ca26718c-Abstract-Conference.html.

Meta AI. Llama-3.3-70B-Instruct model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct.

Maxim Milakov and Natalia Gimelshein. Online normalizer calculation for softmax. *arXiv preprint arXiv:1805.02867*, 2018. URL https://arxiv.org/abs/1805.02867.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. doi: 10.52202/079017-2193. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7ede97c3e082c6df10a8d6103a2eebd2-Abstract-Conference.html. arXiv:2407.08608.

Joe Sharratt. ThriftAttention: Selective mixed precision for long-context FP4 attention. *arXiv preprint arXiv:2605.23081*, 2026. URL https://arxiv.org/abs/2605.23081.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. QUEST: Query-aware sparsity for efficient long-context LLM inference. In *International Conference on Machine Learning (ICML)*, 2024. URL https://proceedings.mlr.press/v235/tang24l.html. arXiv:2406.10774.

Tencent Hunyuan Team. Penguin Video Benchmark. HunyuanVideo project dataset, 2025. URL https://github.com/Tencent-Hunyuan/HunyuanVideo/blob/main/assets/PenguinVideoBenchmark.csv.

Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5e5fd18f863cbe6d8ae392a93fd271c9-Abstract-Conference.html. arXiv:2309.17453.

Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, Jiayan Teng, Zhuoyi Yang, Wendi Zheng, Xiao Liu, Dan Zhang, Ming Ding, Xiaohan Zhang, Shiyu Huang, Xiaotao Gu, Minlie Huang, Jie Tang, and Yuxiao Dong. VisionReward: Fine-grained multi-dimensional human preference learning for image and video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11269–11277, 2026. doi: 10.1609/aaai.v40i13.38107. URL https://ojs.aaai.org/index.php/AAAI/article/view/38107. arXiv:2412.21059.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. XAttention: Block sparse attention with antidiagonal scoring. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 69819–69831. PMLR, 2025. URL https://proceedings.mlr.press/v267/xu25ag.html.

June Yong Yang, Byeongwook Kim, Jeongin Bae, Beomseok Kwon, Gunho Park, Eunho Yang, Se Jung Kwon, and Dongsoo Lee. No token left behind: Reliable KV cache compression via importance-aware mixed precision quantization. *arXiv preprint arXiv:2402.18096*, 2024. URL https://arxiv.org/abs/2402.18096.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/8bd5148caced2d73cea7b6961a874a49-Abstract-Conference.html. arXiv:2505.18875.

Jintao Zhang, Jia Wei, Haoxu Wang, Pengle Zhang, Xiaoming Xu, Haofeng Huang, Kai Jiang, Jianfei Chen, and Jun Zhu. SageAttention3: Microscaling FP4 attention for inference and an exploration of 8-bit training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/4db397e0f760cc573c681e81a01a3dba-Abstract-Conference.html.

Jintao Zhang, Jia Wei, Pengle Zhang, Jun Zhu, and Jianfei Chen. SageAttention: Accurate 8-bit attention for plug-and-play inference acceleration. In *International Conference on Learning Representations (ICLR)*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b286c344d38e10d2466c0514b78e2f36-Abstract-Conference.html. arXiv:2410.02367.

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 *International Conference on Machine Learning (ICML)*, 2025c. URL https://proceedings.mlr.press/v267/zhang25ch.html. arXiv:2502.18137.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. In *International Conference on Machine Learning (ICML)*, 2025d. URL https://proceedings.mlr.press/v267/zhang25m.html.

A Analysis of Mixed-Precision Sparse Attention

This appendix develops three analyses from the main text. First, we derive an exact and efficient test for whether reallocating precision and coverage reduces attention-output error, without recomputing unchanged tiles. Second, we analyze the attention weights themselves, separating the error caused by omitted attention mass from the distortion introduced by lower-precision scores. Third, we study the resulting attention output geometrically: sparsity shifts the attention-weighted value centroid, broader coverage can move it back toward dense attention, and our gain–damage decomposition measures whether this benefit outweighs the numerical error introduced by lower precision.

A.1 Exact attention-error update

Attention contributions and notation. Fix one attention head and one query. Let $\mathcal{J}$ contain its allowed key positions, s_j be the attention score for key j, and $v_j \in \mathbb{R}^{d_v}$ its value vector. Dense attention is

$$\pi_j = \frac{e^{s_j}}{\sum_{\ell\in\mathcal{J}} e^{s_\ell}}, \qquad O = \sum_{j\in\mathcal{J}} \pi_j v_j. \tag{3}$$

We write $j \in i$ for an allowed key position in tile i. An allocation X assigns each tile i a precision $X(i) \in \{\mathrm{BF16}, \mathrm{FP8}, \mathrm{FP4}, \mathrm{skip}\}$. Thus, a superscript such as $(X(i))$ selects the representation assigned to tile i; it is not an exponent. At a computing precision p, let $s_j^{(p)}$ and $v_j^{(p)}$ denote the resulting scores and values, including the effects of quantizing the query, key, and value operands. Define

$$w_i^{(p)} = \sum_{j\in i} e^{s_j^{(p)}}, \qquad a_i^{(p)} = \sum_{j\in i} e^{s_j^{(p)}} v_j^{(p)}. \tag{4}$$

Here, $w_i^{(p)}$ is the total unnormalized attention weight contributed by tile i at precision p, while $a_i^{(p)}$ is the corresponding weighted sum of its value vectors. For a skipped tile, we set $w_i^{(\mathrm{skip})} = 0$ and $a_i^{(\mathrm{skip})} = \mathbf{0}$, where $\mathbf{0}$ is a zero vector. Thus, a skipped tile contributes neither to the softmax normalization nor to the weighted-value sum. For each input, we compute these quantities once for every tile and supported precision; evaluating a candidate allocation then only requires selecting and combining the corresponding cached contributions.

Deriving the error state. Let A_X and W_X collect the weighted-value and attention-mass sums:

$$A_X = \sum_i a_i^{(X(i))}, \qquad W_X = \sum_i w_i^{(X(i))} > 0, \qquad \widetilde{O}_X = \frac{A_X}{W_X}. \tag{5}$$

All computed tiles, regardless of precision, share this denominator. Subtracting the dense output gives

$$\widetilde{O}_X - O = \frac{A_X - W_X O}{W_X} = \frac{N_X}{W_X}, \qquad N_X = \sum_i \left(a_i^{(X(i))} - w_i^{(X(i))} O\right). \tag{6}$$

The vector N_X records how the tiles' contributions combine to produce the final output error. For a fixed query, tile i contributes $a_i^{(X(i))} - w_i^{(X(i))} O$: the difference between its actual weighted-value sum and the sum that would match the dense output O at the same weight $w_i^{(X(i))}$. These differences can offset one another. For example, one tile may contribute a positive deviation in a value dimension while another contributes a negative deviation in that same dimension. We therefore sum these vectors into N_X before measuring the error, which is $\|N_X\|_2/W_X$. Summing their individual magnitudes would lose this cancellation. Keeping (N_X, W_X) allows each edit to be evaluated against the combined contribution of all tiles, including the updated softmax normalization.

Deriving the update. Let B be a baseline allocation and M a candidate, and write $\mathcal{E} = \{i : M(i) \neq B(i)\}$ for the edited tiles. Their total changes are

$$\Delta A = \sum_{i\in\mathcal{E}} \left(a_i^{(M(i))} - a_i^{(B(i))}\right), \qquad \Delta W = \sum_{i\in\mathcal{E}} \left(w_i^{(M(i))} - w_i^{(B(i))}\right). \tag{7}$$

Contributions outside $\mathcal{E}$ are unchanged. Hence $A_M = A_B + \Delta A$ and $W_M = W_B + \Delta W$, and

$$\begin{aligned} N_M &= A_B + \Delta A - (W_B + \Delta W)O \\ &= N_B + \Delta A - \Delta W\, O. \end{aligned} \tag{8}$$

Attention divides the weighted-value sum by the total attention mass. Therefore, to reproduce the dense output O with total mass W, the weighted-value sum must equal WO. After the edit, the total mass becomes $W_B + \Delta W$, so the sum needed to reproduce O becomes $(W_B + \Delta W)O$.

The candidate's actual weighted-value sum is $A_B + \Delta A$. Subtracting the sum needed to reproduce O, then dividing by the updated total mass, gives its attention-output error:

$$\begin{aligned} \widetilde{O}_M - O &= \frac{A_B + \Delta A - (W_B + \Delta W)O}{W_B + \Delta W} \\ &= \frac{N_B + \Delta A - \Delta W\, O}{W_B + \Delta W}, \end{aligned} \tag{9}$$

where $N_B = A_B - W_B O$ from equation 6. The update therefore accounts for two changes: ΔA changes the actual weighted-value sum, while $\Delta W\, O$ changes the sum needed to match the dense output. Their difference is the adjustment to the baseline error numerator.

Since both denominators are positive, M improves the query's output if and only if

$$\frac{\|N_B + \Delta A - \Delta W\, O\|_2}{W_B + \Delta W} < \frac{\|N_B\|_2}{W_B}. \tag{10}$$

This test gives the exact change in attention-output error after the edit. It automatically accounts for both the change in softmax normalization and how the edited tile contributions combine with the error already present in the baseline allocation. Therefore, we can evaluate a candidate using only the baseline state and the contributions of the tiles it changes.

What computation is reused. Before evaluating candidates, we compute the dense reference and cache each tile's contributions at every supported precision. We also maintain the current error state (N_X, W_X). When a candidate changes only k tiles, we update this state using those k cached contributions; all unchanged tiles are reused without recomputing their scores or weighted values. For n_q query rows and value dimension d_v, evaluating such an edit costs $O(kn_q d_v)$, rather than recomputing attention over all tiles. The cached contribution table requires $O(n|\mathcal{P}|n_q d_v)$ storage for n tiles and precision set $\mathcal{P}$. When an improving candidate is selected, it becomes the new current allocation, and its updated (N_X, W_X) state is used to evaluate subsequent candidates.

A.2 Attention-distribution preservation

We next ask whether mixed precision better preserves the attention-weight distribution of dense attention. Expanding coverage brings back attention mass that sparse attention omits, while lower precision can perturb the scores of the interactions that are computed. This analysis separates these two effects and asks which one dominates.

The distributions being compared. Fix one query. Let $p_j = \pi_j$ denote the dense post-softmax attention weight of key j, as defined in Eq. equation 3. The weights $\{p_j\}$ form the dense attention distribution. For an allocation X, let S_X denote the set of keys that it computes, and let $p_X(j)$ denote the precision assigned to the tile containing key j. The attention distribution produced by X is

$$q_j^X = \frac{\mathbf{1}\{j \in S_X\} e^{s_j^{(p_X(j))}}}{\sum_{\ell \in S_X} e^{s_\ell^{(p_X(\ell))}}}. \tag{11}$$

Here, $\mathbf{1}\{j \in S_X\}$ is one when key j is computed and zero when it is skipped. Thus, q_j^X is the post-softmax attention weight assigned to key j under allocation X: computed keys receive weights based on their assigned precision, while skipped keys receive zero weight.

We measure the Kullback–Leibler (KL) divergence

$$D_{\mathrm{KL}}(q^X \| p) = \sum_{j \in S_X} q_j^X \log \frac{q_j^X}{p_j}. \tag{12}$$

KL divergence measures how much the attention weights under X differ from the dense attention weights: a smaller value means that X preserves the dense distribution more closely, and a value of zero means that the two distributions are identical.

We use the direction $D_{\mathrm{KL}}(q^X|p)$ because sparse attention assigns zero weight to skipped keys. This direction remains finite even when some keys are skipped. Importantly, skipped interactions are still reflected in the metric: removing them changes the softmax normalization and therefore changes the attention weights assigned to all remaining computed interactions.

Separating omission from redistribution. Let $P_X = \sum_{j \in S_X} p_j$ be the dense attention mass retained by X and $m_X = 1 - P_X$ the omitted mass. Define the dense distribution conditioned on the computed keys by $p_j^{S_X} = p_j / P_X$ for $j \in S_X$ and zero elsewhere. Then

$$D_{\mathrm{KL}}(q^X \| p) = \underbrace{-\log(1 - m_X)}_{\text{omission distortion}} + \underbrace{D_{\mathrm{KL}}(q^X \| p^{S_X})}_{\text{redistribution within computed interactions}} . \tag{13}$$

To derive the identity, use $p_j = P_X p_j^{S_X}$ on S_X:

$$\begin{aligned} \sum_{j \in S_X} q_j^X \log \frac{q_j^X}{p_j} &= \sum_{j \in S_X} q_j^X \left(\log \frac{q_j^X}{p_j^{S_X}} - \log P_X \right) \\ &= D_{\mathrm{KL}}(q^X \| p^{S_X}) - \log P_X , \end{aligned} \tag{14}$$

since $\sum_{j \in S_X} q_j^X = 1$.

The decomposition has a simple interpretation. The first term measures the error caused by *omitting attention mass*: skipping more important interactions makes this term larger. The second term measures how much the relative attention weights among the computed interactions differ from dense attention. If the computed interactions keep their original scores, this second term is zero. Thus, expanding coverage reduces the omission term, while lower-precision computation may increase the redistribution term. Mixed precision improves the overall attention distribution when the benefit from recovering omitted attention mass outweighs the distortion introduced by lower precision.

A.3 Coverage gain and quantization damage

The KL analysis shows that broader coverage can better preserve the dense attention-weight distribution. We now ask how this affects the attention output itself. Because attention output is a weighted average of value vectors, skipping interactions can shift the output away from its dense-attention value centroid, while recovering useful interactions can move it back. Mixed precision introduces a competing effect: the lower precision that funds this additional coverage can itself perturb the output. We first derive how omission and recovery change the attention output, then use this analysis to motivate the *coverage gain* and *quantization damage* quantities reported in the main text, which directly compare the benefit of broader coverage against the error introduced by lower precision.

How sparsity shifts the value centroid. Attention output is a weighted average of the value vectors. To see what sparsity changes, divide the keys into the selected set S and the omitted set $D = \mathcal{J} \setminus S$. Let

$$m = \sum_{j \in D} p_j$$

be the total dense attention mass assigned to the omitted keys. We can summarize the selected and omitted groups by their attention-weighted value centroids:

$$\mu_S = \frac{\sum_{j \in S} p_j v_j}{1 - m}, \qquad \mu_D = \frac{\sum_{j \in D} p_j v_j}{m}. \tag{15}$$

The sparse output uses only the selected group, so $O_S = \mu_S$. Dense attention, in contrast, combines both groups according to their attention mass:

$$O = (1 - m)\mu_S + m\mu_D. \tag{16}$$

Subtracting the two gives

$$O_S - O = m(\mu_S - \mu_D),$$

and therefore

$$\|O_S - O\|_2 = m\|\mu_S - \mu_D\|_2.$$

This identity makes the effect of sparsity explicit. Omitting interactions matters for two reasons: how much attention mass is removed, and how different the omitted values are from the values that remain. If the omitted interactions have a centroid close to μ_S, skipping them changes the output little. If their centroid is far from μ_S, sparsity can substantially shift the attention output even when the omitted mass is modest.

How expanded coverage restores information. Mixed precision uses the compute saved from lower-precision interactions to recover some keys that sparse attention originally omitted. Let $R \subseteq D$ denote these recovered interactions. We summarize them by their total unnormalized attention mass

$$Z_R = \sum_{j \in R} e^{s_j}$$

and their attention-weighted value centroid

$$\mu_R = \frac{\sum_{j \in R} e^{s_j} v_j}{Z_R}.$$

Similarly, let $Z_S = \sum_{j \in S} e^{s_j}$ be the attention mass of the original sparse set. After adding the recovered interactions, their fraction of the new total mass is

$$\alpha = \frac{Z_R}{Z_S + Z_R}.$$

The expanded attention output is therefore a weighted average of the original sparse output O_S and the recovered centroid μ_R:

$$O_+ = (1 - \alpha)O_S + \alpha\mu_R. \tag{17}$$

Thus, recovering R moves the sparse output toward the information represented by the recovered interactions. The resulting correction is

$$c = O_+ - O_S = \alpha(\mu_R - O_S). \tag{18}$$

The size and direction of this correction depend both on how much attention mass is recovered and on how different the recovered values are from the original sparse output. If all omitted interactions are recovered at their original precision, then $O_+ = O$.

Recovering additional interactions does not automatically guarantee a better output. Let

$$e = O - O_S$$

be the direction from the sparse output toward dense attention. After recovery, the remaining error is $e - c$. Hence,

$$\|O_+ - O\|_2^2 - \|O_S - O\|_2^2 = \|c\|_2^2 - 2\langle e, c\rangle. \tag{19}$$

Recovery improves the output when the correction c points sufficiently toward the dense output. Intuitively, useful recovered interactions move O_S in the right direction; interactions whose values pull the output elsewhere may not help. We therefore measure the benefit of the interactions actually recovered by our mixed-precision allocation rather than assuming that additional coverage is always beneficial.

Separating coverage gain and quantization damage. The centroid analysis explains why recovering omitted interactions can reduce attention-output error: broader coverage can move the sparse output back toward the dense-attention output. In our mixed-precision allocation, however, this extra coverage is funded by lowering the precision of other interactions. We therefore separate the benefit of changing coverage from the error introduced by changing precision.

We compare three outputs from the same attention input. Let O_S be the original sparse output. Next, construct O_Q by keeping the original sparse coverage fixed but applying the precision changes made

by the mixed allocation; this isolates the effect of lower precision. Finally, let O_M be the output of the complete mixed allocation, which also changes which interactions are computed. Thus, the transition

$$O_S \to O_Q \to O_M$$

first changes *precision* and then changes *coverage*.

Using dense attention O as the reference, define

$$E(Y) = \|Y - O\|_F.$$

We then measure

$$\text{Damage} = \frac{E(O_Q) - E(O_S)}{\|O_S\|_F}, \qquad \text{Gain} = \frac{E(O_Q) - E(O_M)}{\|O_S\|_F}. \tag{20}$$

Damage measures how much attention-output error changes when we lower precision while keeping the original sparse coverage fixed. *Gain* measures how much error is subsequently reduced when we change the coverage to obtain the final mixed allocation. Because recovered interactions are evaluated at their actual assigned precisions, their quantization effects are already included in this second step.

Combining the two terms gives

$$\text{Gain} - \text{Damage} = \frac{E(O_S) - E(O_M)}{\|O_S\|_F}. \tag{21}$$

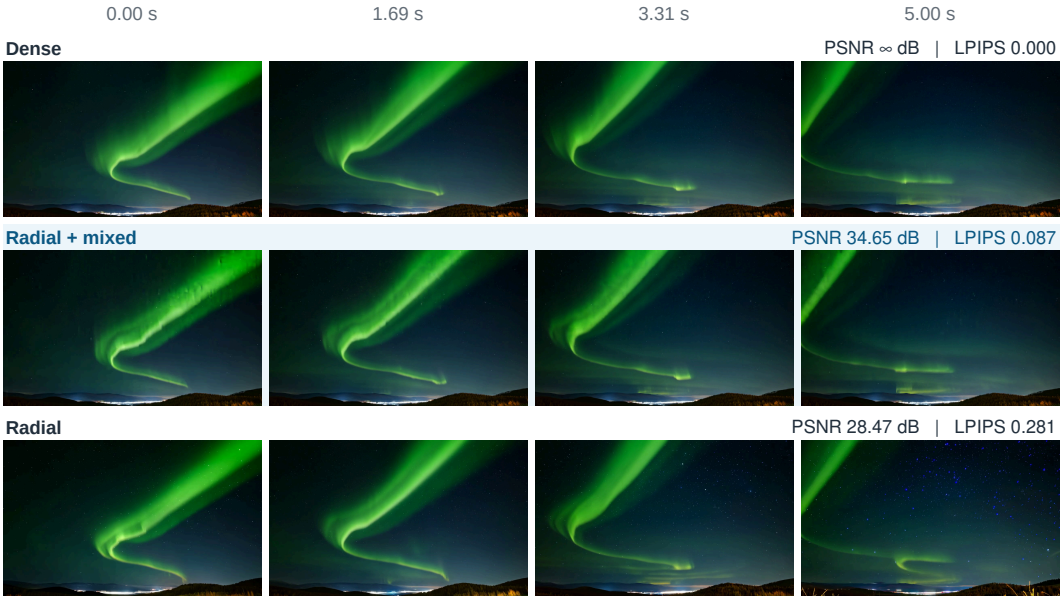
Therefore, $\text{Gain} > \text{Damage}$ means that the final mixed allocation has lower attention-output error than the original sparse allocation. In other words, the benefit obtained from changing coverage outweighs the numerical cost of lowering precision.

B QUALITATIVE VIDEO COMPARISONS

We show two examples for each model–selector pair, chosen from the cases with the largest improvements in full-video LPIPS. Each example shows four frames at matching playback times. From top to bottom, the rows show dense attention, mixed-precision sparse attention, and the original sparse method. Full-video PSNR (higher is better) and LPIPS (lower is better) measure agreement with dense generation. The timestamps indicate playback position, not generation time.

B.1 WAN2.1-14B WITH RADIAL ATTENTION ($\beta = 0.75$)

(a) Prompt: The aurora shines in the sky.



With Radial Attention, the stars become brighter across the shown frames and appear blue in the final frame. The dense and mixed-precision outputs retain a dimmer star field.

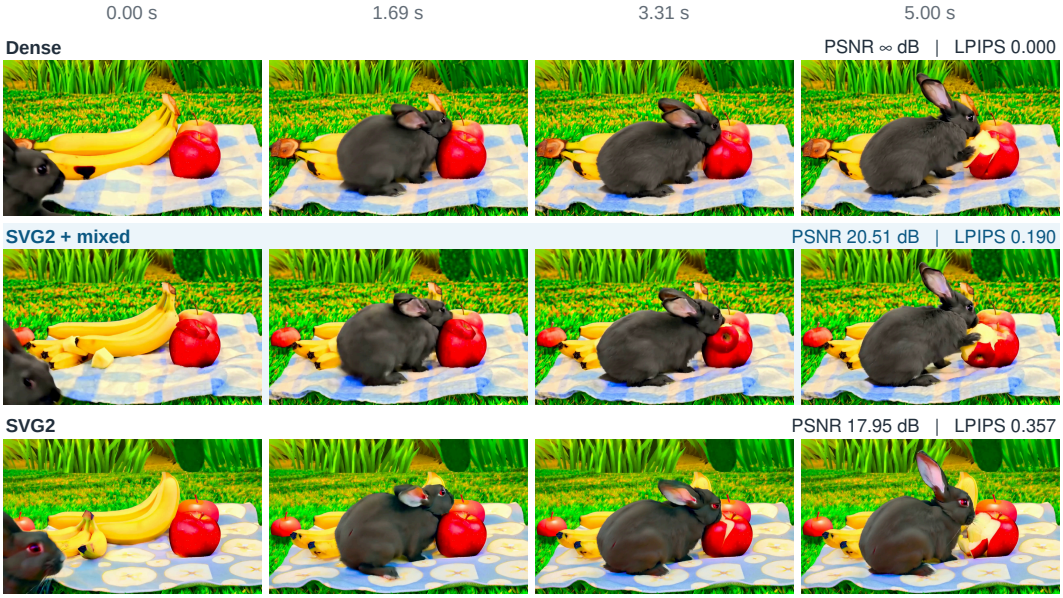
(b) Prompt: In the desolate alien mining site, huge excavation machinery and transport vehicles are scattered around the vast mine pits. Some small reconnaissance machines are flying around in the air.



With Radial Attention, the excavation machinery and the edges of the mine pit change shape and position across frames. These structures remain stable in the dense and mixed-precision outputs.

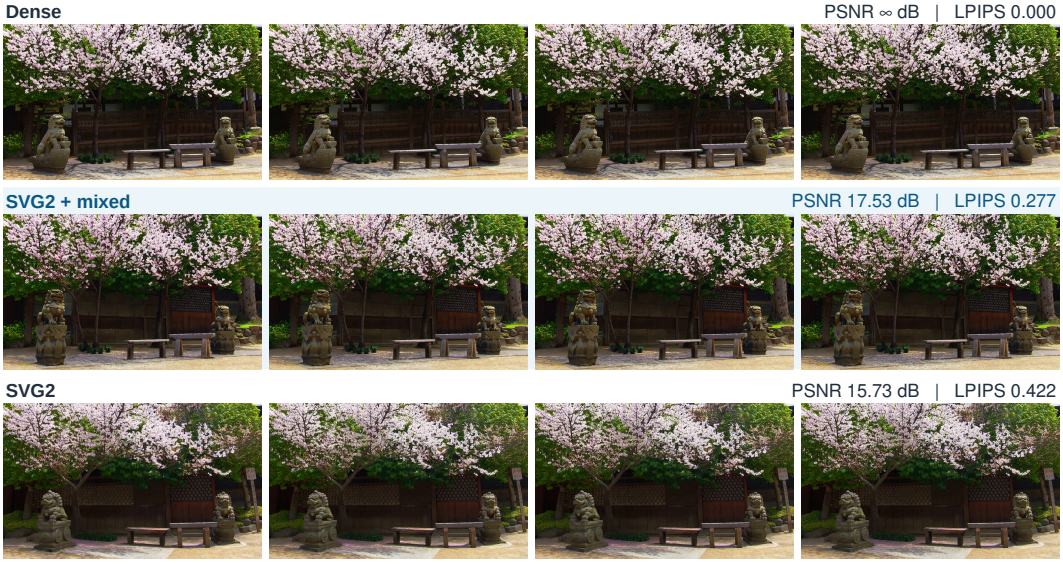
B.2 WAN2.1-14B WITH SVG2 (TOP-p = 0.70)

(a) Prompt: On the grass, a picnic blanket is spread with bananas and apples. A rabbit hops onto the blanket and starts nibbling on an apple. The camera zooms out, revealing a 2K high-definition scene.



SVG2 gives the rabbit red eyes and ears, whereas the dense and mixed-precision outputs show dark eyes and normal ears.

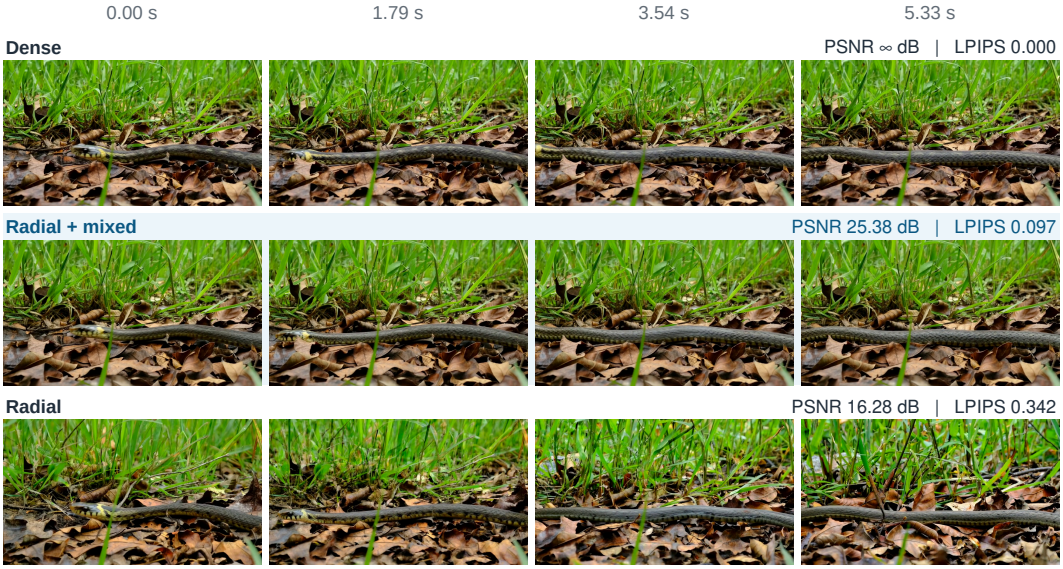
(b) Prompt: Under the cherry blossom trees in Kyoto, there are two stone lions and two wooden stools. A gust of wind blew, and the cherry blossom petals fell one after another.



SVG2 changes the stone lions' orientations and the positions of the wooden stools relative to the dense reference. The mixed-precision output more closely preserves their arrangement.

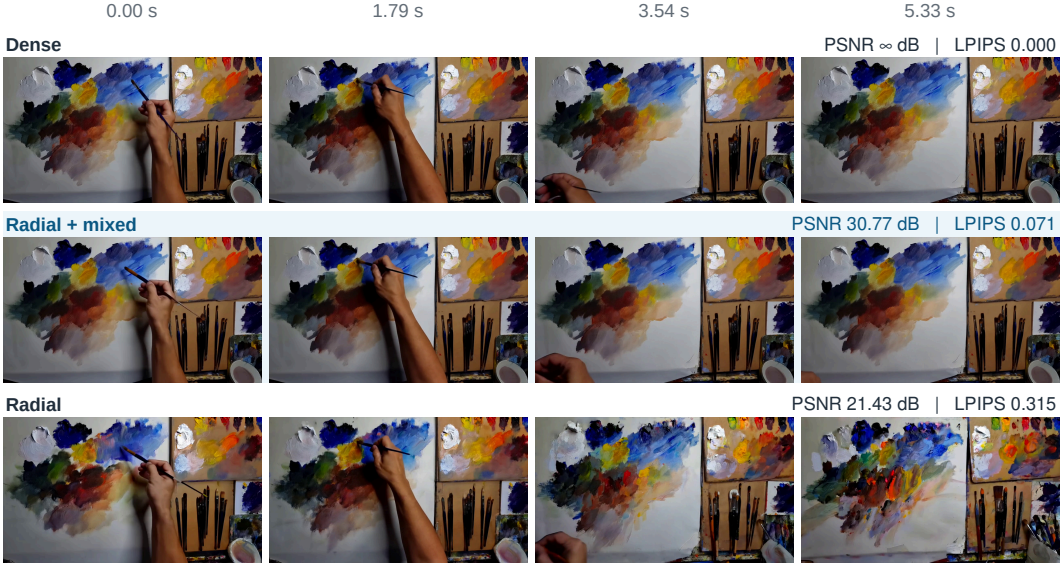
B.3 HUNYUANVIDEO-13B WITH RADIAL ATTENTION ($\beta = 0.75$)

(a) Prompt: A snake slithers smoothly along the ground, weaving agilely through the grass and fallen leaves, tracking shot.



The grass and fallen leaves shift and change shape more visibly in the Radial Attention sequence. Their arrangement remains closer to the dense reference with mixed precision.

(c) Prompt: The brushes lie across the palette, and the palette is placed on the art table, on the right side of the art table are paints, as the camera horizontally moving from left to right.



The white and red paint regions change shape across the Radial Attention frames. Their shapes and locations remain closer to the dense reference in the mixed-precision output.

B.4 HUNYUANVIDEO-13B WITH SVG2 (TOP-$p = 0.70$)

(b) Prompt: In the cold Arctic, the night sky is filled with dazzling auroras that dance like a dreamy scroll, their colorful lights twinkling in the darkness. The camera switches to a cozy cabin, where a man is sitting by the fireplace, smoking a pipe.



SVG2 shows a puff of pipe smoke in the first two frames that is not visible in the corresponding dense and mixed-precision frames.

(d) Prompt: Mickey Mouse from Disney clad in a suit, is conducting the orchestra on stage with elegant hand gestures.



From the second displayed frame onward, Mickey's conducting baton is no longer visible in the SVG2 output, while it remains visible in the dense and mixed-precision outputs.