GRID COMPRESSED ATTENTION: ACCELERATING VIDEO GENERATION ATTENTION WITH TRAINING-FREE GRID KV COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Attention computation is a major bottleneck in video Diffusion Transformers because its cost grows quadratically with the sequence length. Existing training-free sparse attention methods reduce computation via sparse token selection, but the more evenly distributed attention mass in video generation limits how aggressively tokens can be discarded without losing information. We introduce Grid Compressed Attention (GCA), a training-free method that compresses attention via KV compression and query grouping. Specifically, we formulate attention compression as an entropy-regularized optimization problem and derive an efficient construction based on compact KV compressions and disjoint query groups. A local Fisher characterization of the group-logit approximation error guides the selection of representative queries. Experiments on Wan2.1-14B show favorable fidelity–latency trade-offs for GCA among the evaluated configurations. These results support query-guided KV compression as a promising training-free approach to accelerating video generation attention.

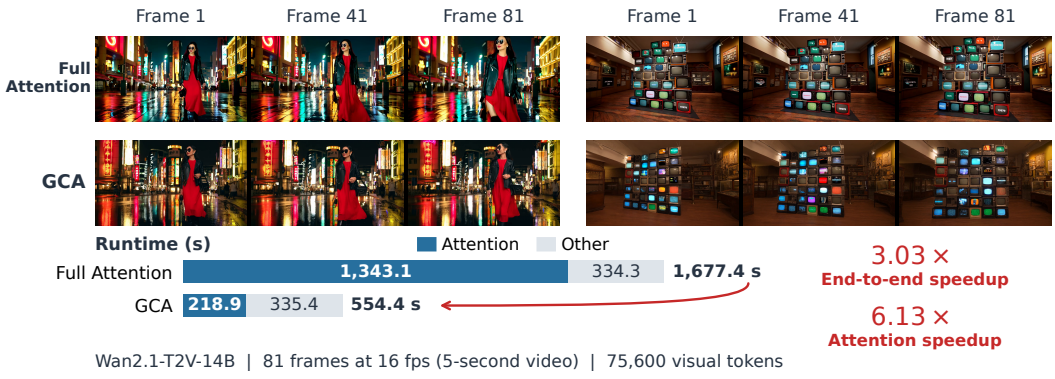


Figure 1: Efficient video generation by GCA with Wan2.1-T2V-14B. The measured runtimes indicate approximately $3.03\times$ end-to-end and $6.13\times$ attention speedups.

1 INTRODUCTION

Video generation models based on Diffusion Transformers (DiTs) produce high-quality videos (Brooks et al., 2024; Peebles & Xie, 2023; Wan et al., 2025; Kong et al., 2024). However, scaled dot-product attention (SDPA) (Vaswani et al., 2017) in DiTs incurs a quadratic computational cost with sequence length. As shown in Figure 1, even a few seconds of high-resolution video can contain hundreds of thousands of visual tokens, and attention becomes a major inference bottleneck. This bottleneck is particularly severe for long videos, where attention can account for the vast majority of inference time (Xi et al., 2025b).

Recent work has explored various approaches to accelerating video attention, with training-free methods particularly appealing for their plug-and-play nature. Sparse attention (Xi et al., 2025a; Yang et al., 2026; Zhou et al., 2026) is a widely used training-free approach that approximates

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.

full attention by predicting and selecting regions with high attention mass for exact computation. Although this approach can achieve high accuracy, its achievable speedup remains limited. Another class of training-free methods uses structured attention, which approximates full attention through structured matrix factorizations but may require multiple iterations or face quality degeneration (Yaras et al., 2026; Liang et al., 2026a; Agarwal et al., 2026). Beyond training-free approaches, linear attention (Katharopoulos et al., 2020), state space models (SSMs) (Gao et al., 2024), and compressed attention (Xu et al., 2026a), which have been widely adopted in LLM inference (Yang et al., 2025b), have also been explored for efficient video generation (Chen et al., 2026a). However, these approaches typically require additional training and cannot be directly applied to existing pretrained models. Developing a training-free attention method that achieves both substantial speedups and superior quality therefore remains an open challenge.

Compared with language models, video DiTs' attention mass scatters more evenly across visual tokens. As shown in Figure 2, the language model (Yang et al., 2025a) recovers 95% of its attention mass using only the top 9% of keys, whereas the video generation model (Wan et al., 2025) requires the top 47%. When attention mass is more evenly distributed, tokens with individually small weights can still make substantial contributions collectively. The token selection therefore faces a dilemma – selecting fewer tokens risks discarding too much attention mass, while retaining more tokens limits the speedup capacity. This motivates us to explore whether we can reduce computation while still preserving as much information as possible from the full sequence.

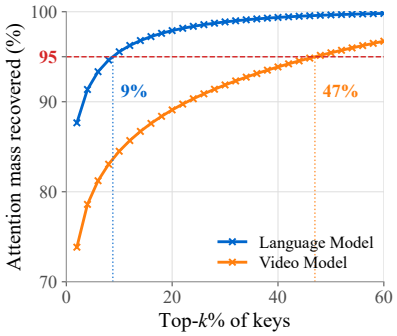


Figure 2: Attention recall of a language model (Qwen3-8B) and a video DiT (Wan2.1-1.3B). Recovering 95% of the attention mass requires the top 9% and 47% of keys, respectively.

In this paper we explore training-free compressed attention for faster video DiT attention. The key idea is to compress multiple key-value (KV) pairs into a compact representative. Specifically, we consider a simple yet effective approach in which each representative is formed as a linear combination of the original KV pairs. By computing attention over this shorter sequence of representatives, each query can attend fewer keys. However, applying compressed attention in a training-free setting is challenging, since we need to determine compression weights and attention weights during inference.

Inspired by previous work (Niculae & Blondel, 2017; Peters et al., 2019; Yaras et al., 2026; Agarwal et al., 2026) that formulates softmax as an entropy-regularized optimization problem, we formulate compressed attention as an entropy-regularized optimization problem over a constrained family of attention distributions induced by linear KV compression. Notably, this formulation is equivalent to minimizing the average reverse KL divergence to the original full attention distributions. Our formulation yields two key observations. First, with an additional constraint on the compression process, the optimal attention weights over compressed tokens have a closed-form solution for a given query. Second, the optimal compression weights vary over the queries, but it will be too expensive to maintain a different set of weights for different queries.

Building on these observations, we introduce Grid Compressed Attention (GCA), a training-free method for accelerating attention in video DiTs. GCA consists of four steps: **1. Grid Partitioning.** GCA partitions queries into groups and keys into blocks, with each pair forming a grid cell. **2. Representative Query Sampling.** Within each cell, GCA samples a small set of queries to guide compression of that key block. **3. Key-Value Compression.** Each key is assigned to the representative query with the largest normalized response. These responses are renormalized over the assigned keys to form compression weights. **4. Entropy-Biased Attention.** Each query performs attention with entropy-bias on its compressed key-values from all key blocks .

Experiments on Wan2.1-14B show that GCA achieves up to $6.2\times$ attention speedups and $3\times$ end-to-end speedups over the full-attention reference on text-to-video (T2V) and image-to-video (I2V) generation. At approximately threefold attention speedup, GCA achieves PSNR above 14 dB on T2V and 16 dB on I2V relative to the corresponding full-attention outputs. Among the evaluated configurations, GCA lies on the Pareto frontier at attention speedups exceeding threefold, demonstrating a favorable balance between generation fidelity and efficiency.

2 PROBLEM FORMULATION

In this section, we formulate KV compression as an optimization problem. Section 2.1 introduces scaled dot-product attention, and Section 2.2 shows that softmax is the unique maximizer of an entropy-regularized objective whose optimality gap is a KL divergence. Section 2.3 expresses attention over linearly compressed key-values with the same objective, so that optimizing the compression minimizes the reverse-KL error to exact attention.

2.1 SCALED DOT-PRODUCT ATTENTION

In video DiTs, the scaled dot-product attention takes the majority of time. Let $\boldsymbol{Q} \in \mathbb{R}^{N\times d}$ be the query matrix and $\boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{N\times d}$ be the key and value matrices, respectively[1], scaled dot-product attention (SDPA) is defined as:

$$\mathrm{SDPA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}_{\mathrm{row}}\left(\boldsymbol{Q}\boldsymbol{K}^\top\right)\boldsymbol{V}. \tag{1}$$

Specifically, for query $\boldsymbol{q}_i$, its attention weight a_{ij} to the key $\boldsymbol{k}_j$ is computed as:

$$z_{ij} = \boldsymbol{q}_i^\top \boldsymbol{k}_j, \qquad \boldsymbol{a}_i = \mathrm{softmax}(\boldsymbol{z}_i), \qquad a_{ij} = \frac{\exp(z_{ij})}{\sum_{\ell=1}^{N}\exp(z_{i\ell})}. \tag{2}$$

2.2 SOFTMAX AS ENTROPY-REGULARIZED OPTIMIZATION

The above mentioned softmax can also be formulated as an entropy-regularized optimization problem (Niculae & Blondel, 2017; Peters et al., 2019; Yaras et al., 2026). Consider the following constrained entropy-regularized objective

$$\mathcal{J}(\boldsymbol{a}_i; \boldsymbol{z}_i) := \boldsymbol{a}_i^\top \boldsymbol{z}_i + \mathcal{H}(\boldsymbol{a}_i), \qquad \boldsymbol{a}_i \in \Delta_N, \tag{3}$$

where $\boldsymbol{z}_i \in \mathbb{R}^N$ is a score vector, $\Delta_N = \{\boldsymbol{a}_i \in \mathbb{R}^N_{\geq 0} : \sum_j a_{ij} = 1\}$ is the probability simplex, and $\mathcal{H}(\boldsymbol{a}_i) = -\sum_j a_{ij}\log a_{ij}$ is the Shannon entropy, using natural logarithms and $0\log 0 = 0$. Maximizing this objective yields the unique closed-form solution, which is exactly the softmax:

$$a^\star_{ij} = \frac{\exp(z_{ij})}{\sum_{\ell=1}^{N}\exp(z_{i\ell})}. \tag{4}$$

For fixed $\boldsymbol{z}_i$ and any $\boldsymbol{a}_i \in \Delta_N$, maximizing $\mathcal{J}(\boldsymbol{a}_i; \boldsymbol{z}_i)$ is equivalent to minimizing the KL divergence from $\boldsymbol{a}_i$ to $\mathrm{softmax}(\boldsymbol{z}_i)$.

$$\mathcal{J}(\boldsymbol{a}^\star_i; \boldsymbol{z}_i) - \mathcal{J}(\boldsymbol{a}_i; \boldsymbol{z}_i) = \mathrm{KL}\left(\boldsymbol{a}_i \,\|\, \mathrm{softmax}(\boldsymbol{z}_i)\right) \geq 0. \tag{5}$$

The derivation and proof for Section 2.2 can be found in Vandenberghe & Boyd (2004) and Wainwright et al. (2008) and we also provide a simpler version in Appendix C.

2.3 LINEAR KV COMPRESSION AS ANOTHER ENTROPY-REGULARIZED OPTIMIZATION

When attention mass in video DiTs is broadly distributed, compression offers a way to aggregate information from many tokens through fewer compressed key-values. Since attention outputs are weighted sums of values, we consider linear combinations to construct these compressed key-values.

Linear KV Compression Consider B_k key-value pairs to be compressed for B_q query vectors. We compress them by a linear combination with nonnegative and normalized weights $\boldsymbol{w}_c = (w_{c1}, \ldots, w_{cB_k})^\top$ into C compressed key/values:

$$\bar{\boldsymbol{k}}_c = \sum_{\ell=1}^{B_k} w_{c\ell}\boldsymbol{k}_\ell, \qquad \bar{\boldsymbol{v}}_c = \sum_{\ell=1}^{B_k} w_{c\ell}\boldsymbol{v}_\ell, \qquad c = 1, \ldots, C. \tag{6}$$

The same weights compress keys and values: $w_{c\ell} \geq 0$ and $\sum_{\ell=1}^{B_k} w_{c\ell} = 1$. Throughout the paper, lowercase c indexes compressed key-value pairs and uppercase C denotes their number within one key block.

[1]For notational simplicity, we absorb the factor $1/\sqrt{d}$ into the keys throughout the paper.

Effective Attention Weight Given a query vector $\boldsymbol{q}_i$, if we compute a weighted sum of these compressed key-value pairs with weight $\boldsymbol{\beta}_i = (\beta_{i1}, ..., \beta_{iC})$, similar to attention, since they are obtained through a linear combination, we can reallocate the weights applied to the compressed value vectors back to the original value vectors.

$$\boldsymbol{o}_i = \sum_{c=1}^{C} \beta_{ic} \bar{\boldsymbol{v}}_c = \sum_{\ell=1}^{B_k} \sum_{c=1}^{C} \beta_{ic} w_{c\ell} \boldsymbol{v}_\ell. \tag{7}$$

We denote the effective weight on original token ℓ for query $\boldsymbol{q}_i$ by $a_{i\ell} = \sum_{c=1}^{C} \beta_{ic} w_{c\ell}$, and collect these weights in the vector $\boldsymbol{a}_i = (a_{i1}, \ldots, a_{iB_k})^\top = \sum_{c=1}^{C} \beta_{ic} \boldsymbol{w}_c$.

Entropy-Regularized Optimization For B_q query vectors, substituting the effective weights $a_{i\ell}$ from Equation 7 into Equation 3 and averaging over the queries yields

$$\mathcal{J} = \frac{1}{B_q} \sum_{i=1}^{B_q} \left[\sum_{\ell=1}^{B_k} a_{i\ell} \boldsymbol{q}_i^\top \boldsymbol{k}_\ell + \mathcal{H}(\boldsymbol{a}_i) \right] = \frac{1}{B_q} \sum_{i=1}^{B_q} \left[\sum_{c=1}^{C} \beta_{ic} \boldsymbol{q}_i^\top \bar{\boldsymbol{k}}_c + \mathcal{H}(\boldsymbol{a}_i) \right]. \tag{8}$$

We maximize $\mathcal{J}$ over $\boldsymbol{\beta}_i \in \Delta_C$ for $i = 1, \ldots, B_q$ and $\boldsymbol{w}_c \in \Delta_{B_k}$ for $c = 1, \ldots, C$. By Equation 5, this is equivalent to minimizing the reverse-KL error relative to the original softmax distributions.

3 METHODOLOGY

In this section, we develop an efficient KV compression method based on the objective in Equation 8. Section 3.1 shows that disjoint merge groups yield a closed-form solution for the attention weights with an entropy correction. Section 3.2 derives query-guided merge weights and introduces representative queries for efficient compression.

3.1 DISJOINT GROUPING FOR CLOSED-FORM ATTENTION

Consider the entropy term in Equation 8 for query $\boldsymbol{q}_i$:

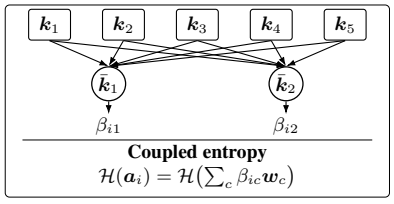
$$\mathcal{H}(\boldsymbol{a}_i) = \mathcal{H}\left(\sum_c \beta_{ic} \boldsymbol{w}_c \right). \tag{9}$$

As shown in Figure 3, when merge supports overlap, this term couples the group weights through shared keys. If the supports are disjoint, each key belongs to only one group and its effective weight is $a_{i\ell} = \beta_{ic} w_{c\ell}$. The entropy then decomposes as

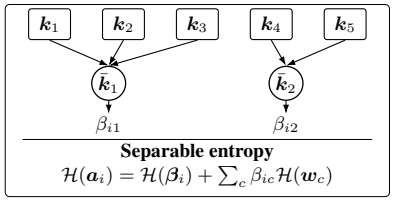
$$\mathcal{H}(\boldsymbol{a}_i) = \mathcal{H}(\boldsymbol{\beta}_i) + \sum_c \beta_{ic} \mathcal{H}(\boldsymbol{w}_c). \tag{10}$$

Disjoint Key Assignment We assign each of the B_k key-value pairs to one representative. For keys not assigned to representative c, set $w_{c\ell} = 0$. Each representative with assigned keys has soft weights that sum to one. For fixed merge weights, substituting Equation 10 into Equation 8 yields an independent problem for each query $\boldsymbol{q}_i$:

$$\max_{\boldsymbol{\beta}_i \in \Delta_C} \left\{ \sum_c \beta_{ic} \boldsymbol{q}_i^\top \bar{\boldsymbol{k}}_c + \sum_c \beta_{ic} \mathcal{H}(\boldsymbol{w}_c) + \mathcal{H}(\boldsymbol{\beta}_i) \right\}. \tag{11}$$



(a) Overlapping supports



(b) Disjoint supports

Figure 3: Overlapping versus disjoint merge supports.

Closed-Form Attention Weights The resulting objective has the same form as the standard softmax problem. Its solution and the attention output are

$$\beta_{ic}^\star = \frac{\exp\left(\boldsymbol{q}_i^\top \bar{\boldsymbol{k}}_c + \mathcal{H}(\boldsymbol{w}_c) \right)}{\sum_{c'=1}^{C} \exp\left(\boldsymbol{q}_i^\top \bar{\boldsymbol{k}}_{c'} + \mathcal{H}(\boldsymbol{w}_{c'}) \right)}, \qquad \boldsymbol{o}_i = \sum_c \beta_{ic}^\star \bar{\boldsymbol{v}}_c. \tag{12}$$

In particular, when no key-value pairs are assigned to representative c, we set its unnormalized weight to zero, giving $\beta^\star_{ic} = 0$ and no contribution to the denominator. This solves $\boldsymbol{\beta}_i$ exactly for the chosen disjoint compression in ideal arithmetic. The remaining task is to assign the B_k key-value pairs and construct their shared merge weights. Appendix D gives the entropy decomposition and both conditional optimization derivations.

3.2 Query-Guided KV Compression

For fixed assignments of the B_k key-value pairs and attention weights $\boldsymbol{\beta}_i$, the optimal merge weights for keys assigned to representative c have a query-key softmax form:

$$\boldsymbol{u}_c = \frac{\sum_i \beta_{ic} \boldsymbol{q}_i}{\sum_i \beta_{ic}}, \qquad w^\star_{c\ell} \propto \exp(\boldsymbol{u}_c^\top \boldsymbol{k}_\ell). \tag{13}$$

Here $\boldsymbol{u}_c$ is the attention-weighted mean query, defined when $\sum_i \beta_{ic} > 0$. Set $w^\star_{c\ell} = 0$ for keys not assigned to representative c, and normalize the remaining weights so that $\sum_{\ell=1}^{B_k} w^\star_{c\ell} = 1$.

To characterize the error from sharing merge weights, fix a nonempty representative c with n assigned keys. Relabel those keys locally as $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_n$ and collect them in $\boldsymbol{K}$. Write

$$\boldsymbol{p}(\boldsymbol{x}) = \mathrm{softmax}(\boldsymbol{K}\boldsymbol{x}), \qquad \boldsymbol{F}(\boldsymbol{x}) = \nabla^2_{\boldsymbol{x}} \log \sum_{j=1}^{n} \exp(\boldsymbol{k}_j^\top \boldsymbol{x}).$$

Here $\boldsymbol{F}(\boldsymbol{x})$ is the Fisher information matrix in query coordinates. When the nonzero entries of $\boldsymbol{w}_c$ equal $\boldsymbol{p}(\boldsymbol{u}_c)$, let $\bar{\boldsymbol{k}}_c$ be the merged key from Equation 6.

Theorem 1 (Group-logit approximation) *For any fixed query $\boldsymbol{u}_c$ and every query $\boldsymbol{q}_i$,*

$$\begin{aligned} d_c(\boldsymbol{q}_i, \boldsymbol{u}_c) &:= \log \sum_{j=1}^{n} \exp(\boldsymbol{k}_j^\top \boldsymbol{q}_i) - \boldsymbol{q}_i^\top \bar{\boldsymbol{k}}_c - \mathcal{H}(\boldsymbol{p}(\boldsymbol{u}_c)) \\ &= \mathrm{KL}(\boldsymbol{p}(\boldsymbol{u}_c) \| \boldsymbol{p}(\boldsymbol{q}_i)) \geq 0, \end{aligned} \tag{14}$$

with equality if and only if $\boldsymbol{p}(\boldsymbol{q}_i) = \boldsymbol{p}(\boldsymbol{u}_c)$. Moreover, as $\boldsymbol{\delta} \to \boldsymbol{0}$,

$$d_c(\boldsymbol{u}_c + \boldsymbol{\delta}, \boldsymbol{u}_c) = \tfrac{1}{2} \boldsymbol{\delta}^\top \boldsymbol{F}(\boldsymbol{u}_c) \boldsymbol{\delta} + O(\|\boldsymbol{\delta}\|^3). \tag{15}$$

The proof is given in Appendix E.

Figure 4 illustrates the query-guided merge construction.

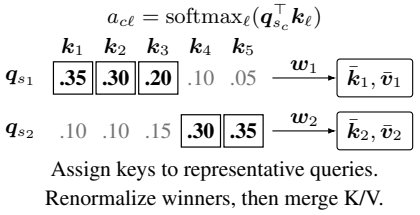


Figure 4: Query-guided KV merging.

Representative Query Sampling Computing the conditional optimum $\boldsymbol{u}_c$ requires $\boldsymbol{\beta}$, which in turn depends on the merge weights. To avoid alternating optimization, we select C representative queries $\boldsymbol{q}_{s_1}, \ldots, \boldsymbol{q}_{s_C}$ from the group and directly set $\boldsymbol{u}_c \leftarrow \boldsymbol{q}_{s_c}$ when constructing the merge weights. This substitution gives a single-pass approximation to the conditional optimization. For a fixed assigned key subset, Theorem 1 characterizes the local group-logit error by a Fisher quadratic form. This motivates using Fisher geometry to select representative queries with complementary attention responses. We construct a probe-based surrogate using P keys from the block and evaluate only R candidate queries. With $\widehat{\boldsymbol{z}}_i$ the probe logits of candidate i, $\boldsymbol{\rho}$ the softmax of their mean, and $\boldsymbol{d}_i = \widehat{\boldsymbol{z}}_i - \widehat{\boldsymbol{z}}_{s_1}$ relative to an anchor s_1 at the spatial position of the key block, the Fisher feature

$$\boldsymbol{\phi}_i = \sqrt{\boldsymbol{\rho}} \odot \left[\boldsymbol{d}_i - (\boldsymbol{\rho}^\top \boldsymbol{d}_i)\mathbf{1}\right] \tag{16}$$

has squared norm given by the Fisher quadratic form of the probe-logit difference at the shared reference $\boldsymbol{\rho}$. Starting from the anchor, we greedily add the candidate whose feature is farthest from the span of the selected ones, until C are chosen. This span-based rule is a heuristic for selecting complementary probe responses. Probing, candidate restriction, and reusing the first frame's selection in later frames are runtime reductions; Appendix F gives the details.

Key Assignment We assign each key to the representative query with the largest normalized response, maximizing total retained attention mass:

$$c^\star(\ell) = \arg\max_{1 \le c \le C} \log a_{c\ell}. \tag{17}$$

Merge Weights For each representative c, initialize $w_{c\ell} = a_{c\ell}$ for its assigned keys and $w_{c\ell} = 0$ for all other keys. If at least one key is assigned, normalize the weights:

$$w_{c\ell} \leftarrow \frac{w_{c\ell}}{\sum_{j=1}^{B_k} w_{cj}}. \tag{18}$$

Query-Aware Compression on a Grid The compression above is query-aware: its merge weights depend on the selected representative queries. To adapt compression to different query groups, we partition queries into groups and key-value pairs into blocks, and compress each block separately for each group. This allows representative selection and merge weights to adapt to the queries within each group. Each group/block pair forms a grid cell, which the next section assembles into the full attention.

4 Grid Compressed Attention

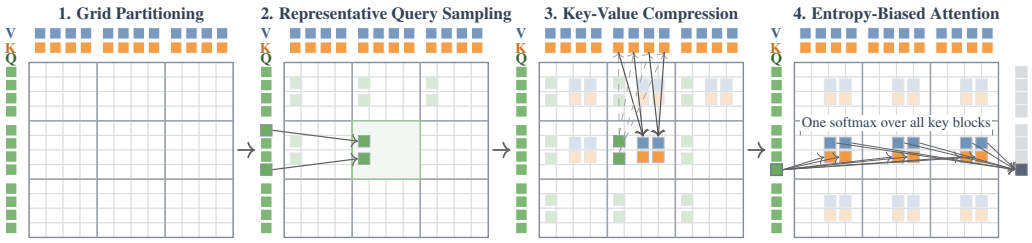


Figure 5: Grid Compressed Attention

In this section, we introduce Grid Compressed Attention (GCA), building on the results obtained in the previous section. As shown in Figure 5, GCA has the following 4 steps:

Step 1: Grid Partitioning. Within each video frame, partition queries into groups of B_q queries and key-value pairs into blocks of B_k pairs. Each query-group/key-block pair forms a grid cell.

Step 2: Representative Query Sampling. Within each cell, select C representatives from its B_q queries: start from a spatially matched anchor, then select queries with complementary responses on a small set of probe keys. Reuse the first query frame's selected indices across frames, loading the corresponding current-frame query vectors.

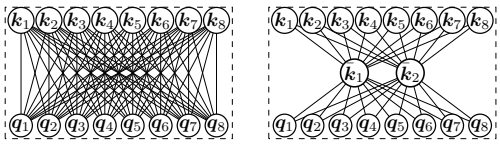
Step 3: Key-Value Compression. Compute each representative query's normalized response over the key block, $a_{c\ell} = \mathrm{softmax}_\ell(\boldsymbol{q}_{s_c}^\top \boldsymbol{k}_\ell)$. Assign each key to the representative with the largest response. Set $w_{c\ell} = a_{c\ell}$ for keys assigned to representative c, and $w_{c\ell} = 0$ otherwise. For representatives with assigned keys, normalize the weights and merge:

$$w_{c\ell} \leftarrow \frac{w_{c\ell}}{\sum_{j=1}^{B_k} w_{cj}}, \qquad \bar{\boldsymbol{k}}_c = \sum_{\ell=1}^{B_k} w_{c\ell} \boldsymbol{k}_\ell, \qquad \bar{\boldsymbol{v}}_c = \sum_{\ell=1}^{B_k} w_{c\ell} \boldsymbol{v}_\ell. \tag{19}$$

Step 4: Entropy-Biased Attention. Note that before Step 4, all operations are performed independently within each grid cell. However, since attention needs to aggregate all key-value pairs for a query, we need to use the compressed key-values from all cells corresponding to that query. Here, we introduce (x, y) to denote the coordinates of the cell formed by the x-th query group and the y-th key group. $\bar{\boldsymbol{k}}_c^{(x,y)}$, $\bar{\boldsymbol{v}}_c^{(x,y)}$, and $\boldsymbol{w}_c^{(x,y)}$ denote the compressed keys, values, and weights obtained in cell (x, y), respectively. For query vector $\boldsymbol{q}_i$, we have:

$$\beta_{ic}^{(x,y)} = \mathrm{softmax}_{y,c}\left(\boldsymbol{q}_i^\top \bar{\boldsymbol{k}}_c^{(x,y)} + \mathcal{H}(\boldsymbol{w}_c^{(x,y)})\right), \qquad \boldsymbol{o}_i = \sum_{y,c} \beta_{ic}^{(x,y)} \bar{\boldsymbol{v}}_c^{(x,y)}. \tag{20}$$

Complexity Analysis. We analyze one attention head with N tokens and head dimension d. For each group of B_q queries, each set of B_k key-value pairs is compressed into C representatives. Assuming full groups and blocks, compression is performed $(N/B_q)(N/B_k) = N^2/(B_qB_k)$ times. As shown in Figure 6, excluding Representative Query Sampling, which is analyzed in Appendix F, Key-Value Compression and Entropy-Biased Attention together have the following cost compared with full attention:



(a) Full Attention: $O(B_qB_kd)$ (b) GCA: $O(CB_qd + CB_kd)$

Figure 6: Computational complexity for B_q queries, B_k key-value pairs, and C compressed key-value pairs.

$$\mathcal{C}_{\text{GCA}} = O\left(N^2 d\left(\frac{C}{B_q} + \frac{C}{B_k}\right)\right) \tag{21}$$

Following Wu et al. (2026a), we align block sizes with the spatial dimensions of video frames for better performance; in our experiments, we use $B_q = 80$ and $B_k = 90$, with C as low as 4.

5 Experiments

5.1 Setup

Models, Datasets, and Metrics. We evaluate on Wan2.1-14B (Wan et al., 2025) T2V and I2V at 720p, using both the 50-step models and their 4-step step-distilled variants (Contributors, 2025), with the 20 prompts of MovieGenVideoBench (Polyak et al., 2025) for T2V and the 20 prompt–image pairs of VBench (Huang et al., 2024) for I2V. Every method replaces all self-attention layers at every step, without full-attention warm-up steps or layers. We report PSNR (Horé & Ziou, 2010), SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) against the full-attention video generated from the same prompt and seed, and VBench scores for video quality. Appendix G gives the arithmetic and memory comparisons.

Baselines. We compare GCA with SVG (Xi et al., 2025a), SVG2 (Yang et al., 2026), SVG-EAR (Zhou et al., 2026), and MonarchRT (Agarwal et al., 2026), using their reference implementations without additional training. Each method has hyperparameters controlling its quality–speed trade-off; we vary these while keeping all other settings at their defaults. We sweep SVG's sparsity over $\{10\%, 15\%, 20\%, 25\%, 30\%\}$ (plus 5% for 50-step T2V and 35% for 4-step T2V); SVG2 and SVG-EAR's top-p thresholds over $\{50\%, 60\%, 70\%, 80\%, 90\%\}$ and $\{40\%, 50\%, 60\%, 70\%, 80\%\}$, respectively; and MonarchRT's row/column reduction factors (h, w) over $\{(1,1), (1,2), (2,1)\}$. We omit configurations with less than $1.5\times$ measured attention speedup over dense attention.

GCA Configuration. In GCA, a query group is one row of a latent frame ($B_q = 80$) and a key block two adjacent columns ($B_k = 90$); no cell spans two frames. We vary the number of representatives $C \in \{4, 6, 8, 10\}$. Representative queries selection is performed on the first latent frame only; the remaining frames reuse the selected positions, taking the queries at the same spatial position in each frame as its representatives.

5.2 Speed–Quality Frontier

In this subsection, we examine whether GCA lies on a better speed–quality frontier than existing methods. Figure 7 plots PSNR and SSIM against per-call attention latency at 50 and 4 steps, and Table 1 compares all methods around a $3\times$ attention speedup.

50 steps. Among the evaluated configurations, GCA provides favorable PSNR–latency and SSIM–latency trade-offs in the regime above $3\times$ attention speedup, while some SVG-EAR settings below $3\times$ attain higher fidelity. The two frontiers cross between GCA's C=8 and C=10 settings, near 115 ms per call on T2V and 120 ms on I2V. On the faster side of the crossing, GCA leads clearly: at about 100 ms per call it is 1–1.8 dB above SVG-EAR and more than 3 dB above SVG and SVG2, and its C=4 setting reaches SVG-EAR's mid-range fidelity at less than half the latency. On the slower side, SVG-EAR's densest setting gains a further 0.5 dB over GCA at the cost of dropping below $3\times$ speedup.

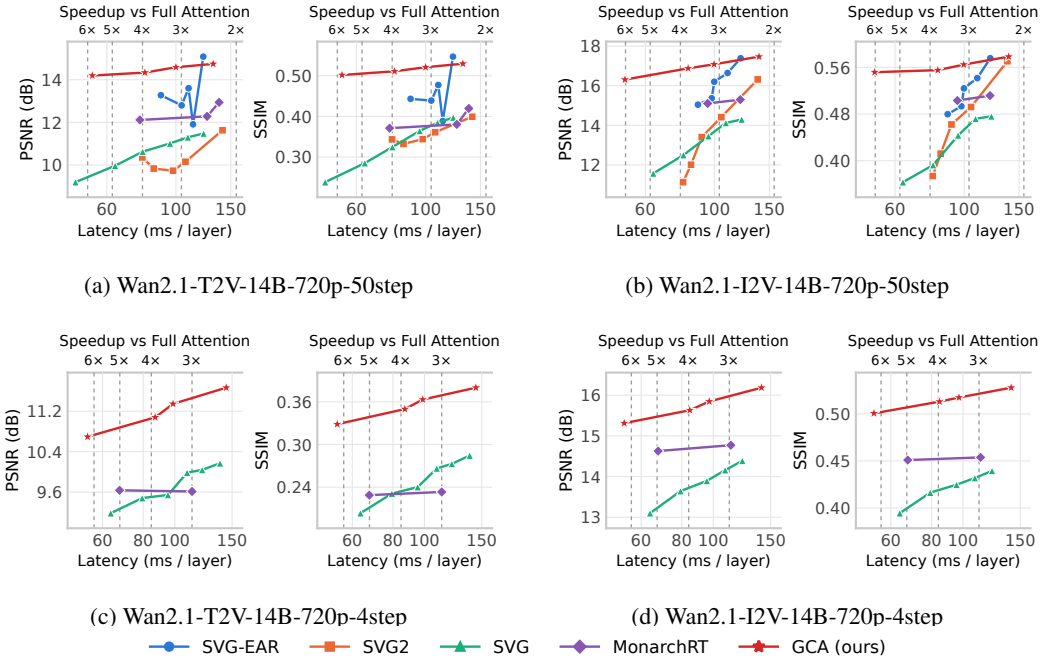


Figure 7: Quality–latency trade-offs for Wan2.1-T2V/I2V-14B at 720p. The 50-step and 4-step groups use the base and step-distilled models, respectively.

Table 1: Comparison on Wan2.1-14B around 3× attention speedup over FlashAttention-2 (Dao, 2024). Each baseline uses its closest setting below 3×, and GCA uses its closest setting above 3×.

	Quality					Efficiency		
Method	PSNR↑	SSIM↑	LPIPS↓	Subj.↑	Img.↑	Attn time↓ (ms/layer)	E2E time↓ (s/video)	Speedup↑
T2V								
Full Attention	–	–	–	95.71	70.88	312.43	1621.0	1.00×
SVG	11.30	0.38	0.59	93.36	69.31	109.18	803.4	2.86×
SVG2	10.15	0.36	0.64	96.86	62.84	107.30	808.3	2.91×
SVG-EAR	12.80	0.44	0.51	95.13	70.44	104.34	783.5	2.99×
MonarchRT	12.28	0.38	0.65	**97.37**	69.44	125.90	806.3	2.48×
GCA	**14.59**	**0.52**	**0.46**	96.80	**71.22**	**100.00**	**743.6**	**3.12**×
I2V								
Full Attention	–	–	–	93.37	71.45	310.91	1624.5	1.00×
SVG	14.12	0.47	0.47	92.37	70.59	108.79	810.4	2.86×
SVG2	14.41	0.49	0.44	**95.05**	70.28	105.19	807.8	2.96×
SVG-EAR	16.64	0.54	**0.36**	91.82	**71.73**	110.17	819.6	2.82×
MonarchRT	15.30	0.51	0.42	94.25	70.20	121.04	858.2	2.57×
GCA	**17.08**	**0.56**	**0.36**	93.26	70.91	**99.70**	**783.9**	**3.12**×

4 steps. In the evaluated four-step configurations, GCA provides better PSNR–latency and SSIM–latency trade-offs than SVG and training-free MonarchRT on both tasks. Its fastest measured setting also has lower attention latency than every evaluated SVG and MonarchRT setting. *Matched speed.* Table 1 compares each baseline at its closest setting below 3× against GCA at its closest setting above 3×. GCA achieves the highest PSNR and SSIM on both tasks: 14.6 dB on T2V and 17.1 dB on I2V, 1.8 and 0.4 dB above SVG-EAR, and at least 2.7 dB above SVG and SVG2, at lower attention latency. GCA has the lowest T2V LPIPS and matches SVG-EAR's lowest I2V LPIPS at 3.12× rather than 2.82× speedup. GCA's VBench subject consistency (96.8 and 93.3) and imaging quality (71.2 and 70.9) remain close to the full-attention reference on both tasks.

Across the evaluated T2V and I2V configurations, these results support favorable PSNR–latency and SSIM–latency trade-offs for GCA under the all-layer replacement protocol.

5.3 Effect of the Number of Representatives

In this subsection, we examine whether fidelity improves as more compressed tokens are retained. The number of representatives C sets the number of compressed key-value pairs per grid cell and is GCA's only accuracy control; Figure 7 shows GCA at $C \in \{4, 6, 8, 10\}$ in all four settings. In every setting, PSNR and SSIM increase monotonically from $C{=}4$ to $C{=}10$ without plateauing in the measured range: PSNR rises by 0.5 dB on 50-step T2V, 1.2 dB on 50-step I2V, 1.0 dB on 4-step T2V, and 0.9 dB on 4-step I2V, while per-call latency grows from 52–54 ms to 132–144 ms. Each additional representative thus buys fidelity at a predictable latency cost, so C is a single monotone knob between speed and quality.

5.4 Component Ablation

In this subsection, we examine whether each of the three components of GCA, the entropy bias, the query-guided merge weights, and the representative query selection, contributes to fidelity. Table 2 removes them one at a time on Wan2.1-14B T2V at $C{=}4$. **Entropy bias.** Dropping the term $\mathcal{H}(\boldsymbol{w})$ from the compressed logits, so that representatives are scored by their merged key alone, lowers PSNR by 4.6 dB: compression alone still recovers much of the dense output, but the bias is what lets the softmax over representatives approximate the original attention distribution (Section 3.1). **Query-guided merging.** Keeping the assignments but averaging the keys and values of each group with equal weights loses 3.0 dB, indicating that query-guided merge weights improve fidelity relative to uniform merging in this setting. **Representative query selection.** Replacing the Fisher selection with four queries drawn uniformly from the same candidate pool loses 0.2 dB in PSNR, with larger degradations in SSIM and LPIPS, so principled selection provides a gain beyond the other two components. These ablations support the contribution of each component in the evaluated setting, with removal of the entropy bias causing the largest PSNR drop.

Table 2: Component ablations of GCA at $C = 4$.

Variant	PSNR ↑	SSIM ↑	LPIPS ↓
GCA	**14.196**	**0.50**	**0.49**
Without entropy bias	9.641	0.31	0.67
Uniform KV merging	11.233	0.36	1.04
Random representative queries	13.986	0.48	0.58

6 Related Work

Sparse attention is a critical component for long-context modeling, in many modalities such as text and videos (Beltagy et al., 2020; Liu et al., 2024; 2025b; Xu et al., 2026b). Recent sparsity work in LLMs focused on processing only critical tokens (Liu et al., 2025a; Xu et al., 2026a; Xiao et al., 2024; Jiang et al., 2024; Lai et al., 2025), which may not fit video generation since spatiotemporal video features are more evenly distributed than text.

For video generation with DiTs, previous static methods (Xi et al., 2025b; Zhang et al., 2025b; Chen et al., 2026b; Li et al., 2026b; Zhao et al., 2026) exploit structured patterns in videos, which fail to generalize to more diverse cases. Dynamic methods (Chen et al., 2025; Cai et al., 2026; Li et al., 2025; 2026a; Liu et al., 2026; Sun et al., 2026; Xia et al., 2025; Wu et al., 2026a; Zhou et al., 2026) process sparse patterns at runtime, often include an identification step with low computation cost to select salient tokens. Another line of work is trainable sparse attention (Zhang et al., 2025a; Agarwal et al., 2026; Liang et al., 2026b; Wu et al., 2026b; Zhang et al., 2026a;b;c), which achieves strong performance, but often requires large compute resources. In contrast to prior methods that either impose predefined patterns or discard unselected tokens, GCA dynamically compresses all K/V tokens into query-guided representations without additional training.

REFERENCES

Krish Agarwal, Zhuoming Chen, Cheng Luo, Yongqi Chen, Haizhong Zheng, Xun Huang, Atri Rudra, and Beidi Chen. Monarchrt: Efficient attention for real-time video generation. *arXiv preprint arXiv:2602.12271*, 2026.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyan Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, et al. Mixture of contexts for long video generation. In *International Conference on Learning Representations*, volume 2026, pp. 92075–92093, 2026.

Aiyue Chen, Yaofu Liu, Junjian Huang, Guang Lian, Yiwu Yao, Wangli Lan, Jing Lin, Zhixin Ma, and Tingting Zhou. Rainfusion2. 0: Temporal-spatial awareness and hardware-efficient block-wise sparse attention. *arXiv preprint arXiv:2512.24086*, 2025.

Junsong Chen, Yuyang Zhao, Jincheng YU, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Zhou Daquan, Huan Ling, Haozhe Liu, Hongwei Yi, Hao Zhang, Muyang Li, Yukang Chen, Han Cai, Sanja Fidler, Ping Luo, Song Han, and Enze Xie. Sana-video: Efficient video generation with block linear diffusion transformer. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 38973–38997, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/41b93c59da0d0f835907fd661d419db2-Paper-Conference.pdf.

Ruichen Chen, Keith Mills, Liyao Jiang, Chao Gao, and Di Niu. Re-ttention: Ultra sparse visual generation via attention statistical reshape. *Advances in Neural Information Processing Systems*, 38:58029–58055, 2026b.

LightX2V Contributors. Lightx2v: Light video generation inference framework. https://github.com/ModelTC/lightx2v, 2025.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mZn2Xyh9Ec.

Yu Gao, Jiancheng Huang, Xiaopeng Sun, Zequn Jie, Yujie Zhong, and Lin Ma. Matten: Video generation with mamba-attention. *arXiv preprint arXiv:2405.03025*, 2024.

Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pp. 2366–2369, 2010. doi: 10.1109/ICPR.2010.579.

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, June 2024.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H Abdi, Dongsheng Li, Chin-Yew Lin, et al. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *Advances in Neural Information Processing Systems*, 37:52481–52515, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

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.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. Flexprefill: A context-aware sparse attention mechanism for efficient long-sequence inference. *arXiv preprint arXiv:2502.20766*, 2025.

Haopeng Li, Shitong Shao, Wenliang Zhong, Zikai Zhou, Lichen Bai, Hui Xiong, and Zeke Xie. Pisa: Piecewise sparse attention is wiser for efficient diffusion transformers. *arXiv preprint arXiv:2602.01077*, 2026a.

Xiaolong Li, Youping Gu, Xi Lin, Weijie Wang, and Bohan Zhuang. Psa: Pyramid sparse attention for efficient video understanding and generation. *arXiv preprint arXiv:2512.04025*, 2025.

Xingyang Li, Muyang Li, Tianle Cai, Haocheng Xi, Shuo Yang, Yujun Lin, Lvmin Zhang, Songlin Yang, Jinbo Hu, Kelly Peng, et al. Radial attention: O(nlog n) sparse attention with energy decay for long video generation. *Advances in Neural Information Processing Systems*, 38:16822–16852, 2026b.

Cheng Liang, Haoxian Chen, Liang Hou, Qi Fan, Gangshan Wu, Xin Tao, and Limin Wang. Vmonarch: Efficient video diffusion transformers with structured attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4613–4623, June 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/html/Liang_VMonarch_Efficient_Video_Diffusion_Transformers_with_Structured_Attention_CVPR_2026_paper.html.

Cheng Liang, Haoxian Chen, Liang Hou, Qi Fan, Gangshan Wu, Xin Tao, and Limin Wang. Vmonarch: Efficient video diffusion transformers with structured attention. *arXiv preprint arXiv:2601.22275*, 2026b.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Ringattention with blockwise transformers for near-infinite context. In *International Conference on Learning Representations*, volume 2024, pp. 3992–4008, 2024.

Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with blockwise ringattention. In *International Conference on Learning Representations*, volume 2025, pp. 45953–45977, 2025b.

Yuxi Liu, Yipeng Hu, Zekun Zhang, Kunze Jiang, and Kun Yuan. Mixture of distributions matters: Dynamic sparse attention for efficient video diffusion transformers. *arXiv preprint arXiv:2601.11641*, 2026.

Vlad Niculae and Mathieu Blondel. A regularized framework for sparse and structured neural attention. *Advances in neural information processing systems*, 30, 2017.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Ben Peters, Vlad Niculae, and André FT Martins. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1504–1519, 2019.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell,

Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models, 2025. URL https://arxiv.org/abs/2410.13720.

Wenhao Sun, Rong-Cheng Tu, Yifu Ding, Jingyi Liao, Zhao Jin, Shunyu Liu, and Dacheng Tao. Vorta: Efficient video diffusion via routing sparse attention. *Advances in Neural Information Processing Systems*, 38:7837–7863, 2026.

Lieven Vandenberghe and Stephen Boyd. *Convex optimization*, volume 1. Cambridge university press Cambridge, 2004.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Martin J Wainwright, Michael Irwin Jordan, et al. Graphical models, exponential families, and variational inference. 2008.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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. doi: 10.1109/TIP.2003.819861.

Jianzong Wu, Liang Hou, Haotian Yang, Ye Tian, Pengfei Wan, Di ZHANG, and Yunhai Tong. Vmoba: Mixture-of-block attention for video diffusion models. In *International Conference on Learning Representations*, volume 2026, pp. 132534–132548, 2026a.

Xinjian Wu, Hongmei Wang, Yuan Zhou, and Qinglin Lu. Usv: Unified sparsification for accelerating video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4180–4189, 2026b.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68208–68224. PMLR, 13–19 Jul 2025a. URL https://proceedings.mlr.press/v267/xi25c.html.

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*, 2025b.

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

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. Infllm: Training-free long-context extrapolation for llms with an efficient context memory. *Advances in neural information processing systems*, 37:119638–119661, 2024.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026a.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, et al. Sparse videogen2: Accelerate video generation with sparse attention via semantic-aware permutation. *Advances in Neural Information Processing Systems*, 38:96965–96991, 2026.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations*, volume 2025, pp. 29687–29707, 2025b.

Can Yaras, Alec Xu, Pierre Abillama, Changwoo Lee, and Laura Balzano. Monarchattention: Zero-shot conversion to fast, hardware-aware structured attention. *Advances in Neural Information Processing Systems*, 38:3445–3471, 2026.

Jintao Zhang, Kai Jiang, Chendong Xiang, Weiqi Feng, Yuezhou Hu, Haocheng Xi, Jianfei Chen, and Jun Zhu. Spargeattention2: Trainable sparse attention via hybrid top-k+ top-p masking and distillation fine-tuning. *arXiv preprint arXiv:2602.13515*, 2026a.

Jintao Zhang, Haoxu Wang, Kai Jiang, Shuo Yang, Kaiwen Zheng, Haocheng Xi, Ziteng Wang, Hongzhou Zhu, Min Zhao, Ion Stoica, et al. Sla: Beyond sparsity in diffusion transformers via fine-tunable sparse–linear attention. In *International Conference on Learning Representations*, volume 2026, pp. 138288–138305, 2026b.

Jintao Zhang, Haoxu Wang, Kai Jiang, Kaiwen Zheng, Youhe Jiang, Ion Stoica, Jianfei Chen, Jun Zhu, and Joseph E Gonzalez. Sla2: Sparse-linear attention with learnable routing and qat. *arXiv preprint arXiv:2602.12675*, 2026c.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric P. Xing, and Hao Zhang. Faster video diffusion with trainable sparse attention. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=VrYCLQ5inI.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. *arXiv preprint arXiv:2502.04507*, 2025b.

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 (CVPR)*, June 2018.

Tianchen Zhao, Ke Hong, Xinhao Yang, Xuefeng Xiao, Huixia Li, Feng Ling, Ruiqi Xie, Siqi Chen, Hongyu Zhu, Zhang Yichong, et al. Paroattention: Pattern-aware reordering for efficient sparse and quantized attention in visual generation models. *Advances in Neural Information Processing Systems*, 38:126484–126511, 2026.

Xuanyi Zhou, Qiuyang Mang, Shuo Yang, Haocheng Xi, Jintao Zhang, Huanzhi Mao, Joseph E. Gonzalez, Kurt Keutzer, Ion Stoica, and Alvin Cheung. Svg-ear: Parameter-free linear compensation for sparse video generation via error-aware routing. *arXiv preprint arXiv:2603.08982*, 2026.

A USE OF AI TOOLS

We used AI tools to assist with manuscript preparation, including language polishing and improvements to clarity and readability, to assist in the writing of proofs, and to support qualitative and thematic data analysis. We also used AI coding tools to assist with implementing and running experiments and optimizing GPU kernels. The authors take full responsibility for the final manuscript, code, and reported results.

B REPRODUCIBILITY

Our code will be released at `https://anonymous.4open.science/r/GCA-ICLR2027-50A0`.

C PROOF OF THE ENTROPY-REGULARIZED SOFTMAX

Fix a finite score vector $\boldsymbol{z} \in \mathbb{R}^N$, let $Z = \sum_{\ell=1}^{N} \exp(z_\ell)$, and let $\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{z})$, so that $p_j > 0$ and $\log p_j = z_j - \log Z$. For any $\boldsymbol{a} \in \Delta_N$, the divergence $\mathrm{KL}(\boldsymbol{a}\|\boldsymbol{p})$ is finite, and with $0 \log 0 = 0$,

$$\mathrm{KL}(\boldsymbol{a}\|\boldsymbol{p}) = \sum_{j=1}^{N} a_j \log a_j - \sum_{j=1}^{N} a_j z_j + \log Z = \log Z - \mathcal{J}(\boldsymbol{a}; \boldsymbol{z}). \tag{22}$$

Taking $\boldsymbol{a} = \boldsymbol{p}$ gives $\mathcal{J}(\boldsymbol{p}; \boldsymbol{z}) = \log Z$, hence $\mathcal{J}(\boldsymbol{p}; \boldsymbol{z}) - \mathcal{J}(\boldsymbol{a}; \boldsymbol{z}) = \mathrm{KL}(\boldsymbol{a}\|\boldsymbol{p})$. The inequality $-\log t \geq 1 - t$, with equality only at $t = 1$, gives $\mathrm{KL}(\boldsymbol{a}\|\boldsymbol{p}) = -\sum_{j:a_j>0} a_j \log \frac{p_j}{a_j} \geq 1 - \sum_{j:a_j>0} p_j \geq 0$, with equality if and only if $\boldsymbol{a} = \boldsymbol{p}$. Therefore $\boldsymbol{p}$ is the unique maximizer of Equation 3, which proves Equation 4; the optimal value is $\log Z$, and Equation 5 holds. Since $\log Z$ does not depend on $\boldsymbol{a}$, maximizing $\mathcal{J}(\boldsymbol{a}; \boldsymbol{z})$ over any subset of the simplex is equivalent to minimizing $\mathrm{KL}(\boldsymbol{a}\|\,\mathrm{softmax}(\boldsymbol{z}))$ over that subset, which is the observation used in Section 2.3.

D CONDITIONAL OPTIMIZATION WITH DISJOINT COMPRESSION

We work within one grid cell with B_q queries and B_k key-value pairs, and omit its coordinates. Representative $c \in \{1, \ldots, C\}$ has merge weights $w_{c\ell} \geq 0$ with $\sum_\ell w_{c\ell} = 1$, supported on its assigned keys only. Empty representatives carry no attention mass and are omitted.

Entropy decomposition. Because the supports are disjoint, $a_{i\ell} = \beta_{ic} w_{c\ell}$ for the unique representative c owning key ℓ, so

$$\mathcal{H}(\boldsymbol{a}_i) = -\sum_{c=1}^{C} \sum_{\ell} \beta_{ic} w_{c\ell} \log(\beta_{ic} w_{c\ell}) = \mathcal{H}(\boldsymbol{\beta}_i) + \sum_{c=1}^{C} \beta_{ic} \mathcal{H}(\boldsymbol{w}_c). \tag{23}$$

Optimal attention weights. For fixed merge weights, the objective in $\boldsymbol{\beta}_i$ is entropy-regularized and linear with scores $\boldsymbol{q}_i^\top \bar{\boldsymbol{k}}_c + \mathcal{H}(\boldsymbol{w}_c)$. By Appendix C, its unique maximizer is the softmax of these scores over the nonempty representatives, which is Equation 12; empty representatives receive zero unnormalized weight.

Optimal merge weights. For fixed assignments and $\boldsymbol{\beta}_i$, the terms depending on $\boldsymbol{w}_c$ are

$$\sum_{i=1}^{B_q} \beta_{ic} \left[\sum_{\ell} w_{c\ell} \boldsymbol{q}_i^\top \boldsymbol{k}_\ell + \mathcal{H}(\boldsymbol{w}_c) \right] = \left(\sum_{i=1}^{B_q} \beta_{ic} \right) \left[\sum_{\ell} w_{c\ell} \boldsymbol{u}_c^\top \boldsymbol{k}_\ell + \mathcal{H}(\boldsymbol{w}_c) \right], \qquad \boldsymbol{u}_c = \frac{\sum_i \beta_{ic} \boldsymbol{q}_i}{\sum_i \beta_{ic}}. \tag{24}$$

When $\sum_i \beta_{ic} > 0$, Appendix C applied to the assigned keys gives $w^\star_{c\ell} \propto \exp(\boldsymbol{u}_c^\top \boldsymbol{k}_\ell)$ on those keys and zero elsewhere, which is Equation 13. If $\sum_i \beta_{ic} = 0$, any feasible $\boldsymbol{w}_c$ is optimal. In the single-pass algorithm, we replace the conditional optimum $\boldsymbol{u}_c$ by a selected representative query $\boldsymbol{q}_{s_c}$ to construct the merge weights. The assignment rule in Equation 17 maximizes retained response mass key by key and is a surrogate for the joint objective.

E Proof of Theorem 1

Relabel the keys assigned to representative c as $\boldsymbol{K} \in \mathbb{R}^{n\times d}$, and write $Z(\boldsymbol{x}) = \log \sum_{j=1}^{n} \exp(\boldsymbol{k}_j^\top \boldsymbol{x})$, so that $\boldsymbol{p}(\boldsymbol{x}) = \mathrm{softmax}(\boldsymbol{K}\boldsymbol{x})$, $\nabla Z(\boldsymbol{x}) = \boldsymbol{K}^\top \boldsymbol{p}(\boldsymbol{x})$, and $\nabla^2 Z(\boldsymbol{x}) = \boldsymbol{K}^\top[\mathrm{diag}(\boldsymbol{p}(\boldsymbol{x})) - \boldsymbol{p}(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x})^\top]\boldsymbol{K} = \boldsymbol{F}(\boldsymbol{x})$. By Equation 6, $\bar{\boldsymbol{k}}_c = \boldsymbol{K}^\top \boldsymbol{p}(\boldsymbol{u}_c) = \nabla Z(\boldsymbol{u}_c)$, and since $\log p_j(\boldsymbol{u}_c) = \boldsymbol{k}_j^\top \boldsymbol{u}_c - Z(\boldsymbol{u}_c)$,

$$\mathcal{H}(\boldsymbol{p}(\boldsymbol{u}_c)) = Z(\boldsymbol{u}_c) - \boldsymbol{u}_c^\top \bar{\boldsymbol{k}}_c, \qquad d_c(\boldsymbol{q}_i, \boldsymbol{u}_c) = Z(\boldsymbol{q}_i) - Z(\boldsymbol{u}_c) - (\boldsymbol{q}_i - \boldsymbol{u}_c)^\top \bar{\boldsymbol{k}}_c. \tag{25}$$

Expanding the divergence with the same log-probabilities gives

$$\mathrm{KL}(\boldsymbol{p}(\boldsymbol{u}_c)\|\boldsymbol{p}(\boldsymbol{q}_i)) = \sum_{j=1}^{n} p_j(\boldsymbol{u}_c)\big[\boldsymbol{k}_j^\top(\boldsymbol{u}_c - \boldsymbol{q}_i) + Z(\boldsymbol{q}_i) - Z(\boldsymbol{u}_c)\big] = Z(\boldsymbol{q}_i) - Z(\boldsymbol{u}_c) - (\boldsymbol{q}_i - \boldsymbol{u}_c)^\top \bar{\boldsymbol{k}}_c, \tag{26}$$

which equals $d_c(\boldsymbol{q}_i, \boldsymbol{u}_c)$. Nonnegativity and the equality condition follow from Appendix C: $d_c(\boldsymbol{q}_i, \boldsymbol{u}_c) = 0$ exactly when $\boldsymbol{p}(\boldsymbol{q}_i) = \boldsymbol{p}(\boldsymbol{u}_c)$, that is, when $\boldsymbol{K}(\boldsymbol{q}_i - \boldsymbol{u}_c) \in \mathrm{span}\{\mathbf{1}\}$. This proves Equation 14.

Equation 25 shows that $d_c(\boldsymbol{u}_c + \boldsymbol{\delta}, \boldsymbol{u}_c)$ is the remainder of the first-order Taylor expansion of Z at $\boldsymbol{u}_c$. Since Z is smooth with locally bounded third derivatives for fixed finite $\boldsymbol{K}$, Taylor's theorem gives

$$d_c(\boldsymbol{u}_c + \boldsymbol{\delta}, \boldsymbol{u}_c) = \tfrac{1}{2}\boldsymbol{\delta}^\top \boldsymbol{F}(\boldsymbol{u}_c)\boldsymbol{\delta} + O(\|\boldsymbol{\delta}\|^3),$$

which proves Equation 15 without any rank assumption on $\boldsymbol{K}$. ∎

F Representative Query Sampling and Implementation

Probe Fisher metric. With $\boldsymbol{\rho}$ the softmax of the mean probe logits and $\boldsymbol{d}_i = \widehat{\boldsymbol{z}}_i - \widehat{\boldsymbol{z}}_{s_1}$ as in Section 3.2, the feature in Equation 16 satisfies

$$\|\boldsymbol{\phi}_i\|_2^2 = \boldsymbol{d}_i^\top[\mathrm{diag}(\boldsymbol{\rho}) - \boldsymbol{\rho}\boldsymbol{\rho}^\top]\boldsymbol{d}_i, \tag{27}$$

where the bracketed matrix is the softmax Fisher matrix in logit coordinates. Centering removes constant logit shifts, which do not change attention probabilities. With P probe keys this is a sampled metric.

Selection rule. Start with the spatially matched anchor s_1. At each iteration, project every unused candidate feature onto the span of the selected features and choose the candidate with the largest squared residual norm, appending its residual direction to an orthonormal basis. If all residuals vanish, take the first unused candidate in the fixed candidate order. This yields exactly C distinct indices including the anchor. We use $R = 32$ candidates and $P = 16$ uniformly spaced probes, and break ties by candidate order.

Reuse and empty representatives. Selection runs on the first latent query frame, and its spatial indices are reused across query frames; the current-frame query vectors still determine the merge weights at every layer and denoising step. Each key is assigned to exactly one representative. An empty representative keeps its slot and is masked from the final softmax; no original key or residual route is added.

Arithmetic complexity. Counting one multiply and one addition as two FLOPs, one query-group/key-block pair and one head spends $2CB_kd$ FLOPs on representative-to-key scoring, $4CB_kd$ on forming compressed keys and values, and $4B_qCd$ on attention from the queries to the representatives. Summing over $N^2/(B_qB_k)$ pairs gives

$$N^2 d\left(\frac{6C}{B_q} + \frac{4C}{B_k}\right), \tag{28}$$

which implies Equation 21. Exponentiation, normalization, assignment, and entropy accumulation are outside this count but inside the measured latency. With T latent frames and first-frame reuse, there are $N^2/(TB_qB_k)$ selection problems, each costing $2RPd$ FLOPs for candidate/probe products and $O(RPC)$ for the residual updates, so selection costs $O\big(N^2(RPd + RPC)/(TB_qB_k)\big)$.

Numerics. Keys, values, and compressed keys and values are stored in BF16; the entropy term is FP32. The identities above hold in exact arithmetic; floating-point softmax and near-tied assignments may introduce small numerical differences.

G ARITHMETIC AND MEMORY-READ COMPARISONS

The main figures compare quality against latency. Here we separate two further cost axes: arithmetic for GCA and the SVG family, and global-memory reads for GCA and MonarchRT. Neither implies an equal reduction in measured latency.

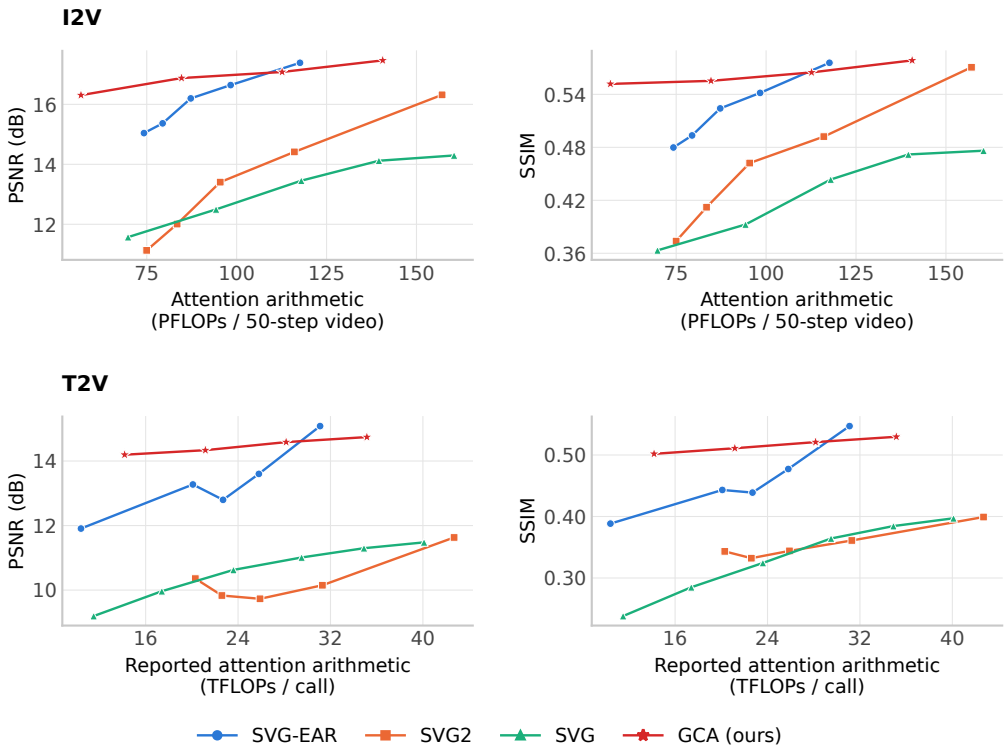


Figure 8: Quality versus attention arithmetic for GCA and the SVG family on 50-step Wan2.1-14B. I2V (top) reports PFLOPs over the full 50-step video; T2V (bottom) reports TFLOPs per attention call. Quality values are the same as in the main latency figure.

Arithmetic versus the SVG family. Figure 8 compares GCA with SVG, SVG2, and SVG-EAR at each method's measured quality. The I2V arithmetic covers all 4,000 calls, including the clustering schedule of adaptive sparse methods; GCA's count includes candidate/probe scoring, representative-to-key scoring, KV compression, and attention over compressed KV. At $C = 4$, GCA uses 56.72 PFLOPs per I2V video and reaches 16.31 dB PSNR and 0.5518 SSIM, while SVG-EAR at $p = 0.6$ uses 87.24 PFLOPs for 16.20 dB and 0.5241, so GCA uses about 35% less arithmetic at similar PSNR. The highest-density settings of both families overlap in quality, so the curves show a trade-off rather than dominance at every budget.

Memory reads versus MonarchRT. Figure 9 counts source-level global-memory load payloads for one complete operator on a 75,600-token sequence with 40 heads and head dimension 128, including repeated loads across thread blocks, representative selection, merge and attention kernels, and layout copies; original and compressed KV are BF16 and entropy is FP32. The counts characterize logical demand on the memory hierarchy: loads may be served by caches, and stores, transaction padding, and register spills are excluded. Over a 50-step video, GCA $C = 4, 6, 8, 10$ reads 482.00, 679.12, 882.43, and 1098.13 TB, and MonarchRT $1/1/1$ and $1/1/2$ reads 732.82 and 948.52 TB. At $C = 4$, GCA has 34.2% fewer modeled reads than MonarchRT $1/1/1$ with 16.31 versus 15.10 dB PSNR and 0.5518 versus 0.5032 SSIM. This does not imply the same saving in physical HBM traffic or runtime, since GCA also materializes compressed KV.

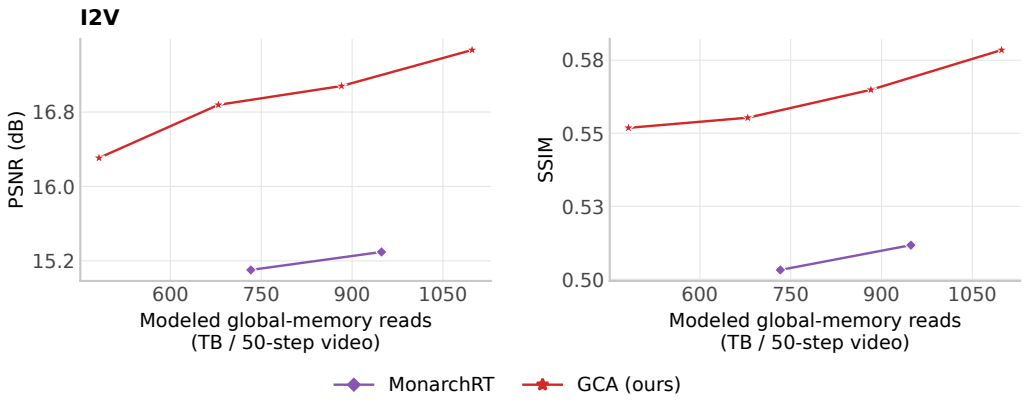


Figure 9: I2V quality versus modeled global-memory reads for GCA and MonarchRT, summed over a full 50-step video. These are analytical load-payload counts, not measured HBM transfers. GCA shows $C = 4, 6, 8, 10$; MonarchRT shows $f/h/w = 1/1/1$ and $1/1/2$.