# SVG-EAR: Parameter-Free Linear Compensation for Sparse Video Generation via Error-aware Routing

Xuanyi Zhou†, Qiuyang Mang†, Shuo Yang, Haocheng Xi,
Jintao Zhang, Huanzhi Mao, Joseph E. Gonzalez,
Kurt Keutzer, Ion Stoica, and Alvin Cheung

UC Berkeley, USA
†Equal contribution.

**Abstract.** Diffusion Transformers (DiTs) have become a leading backbone for video generation, yet their quadratic attention cost remains a major bottleneck. Sparse attention reduces this cost by computing only a subset of attention blocks. However, prior methods often either drop the remaining blocks which incurs information loss, or rely on learned predictors to approximate them, introducing training overhead and potential output distribution shifting. In this paper, we show that the missing contributions can be recovered *without training*: after semantic clustering, keys and values within each block exhibit strong similarity and can be well summarized by a small set of cluster centroids. Based on this observation, we introduce `SVG-EAR`, a parameter-free linear compensation branch that uses the centroid to approximate skipped blocks and recover their contributions. While centroid compensation is accurate for most blocks, it can fail on a small subset. Standard sparsification typically selects blocks by attention scores, which indicate where the model places its attention mass, but not where the approximation error would be largest. `SVG-EAR` therefore performs error-aware routing: a lightweight probe estimates the compensation error for each block, and we compute exactly the blocks with the highest error-to-cost ratio while compensating for skipped blocks. We provide theoretical guarantees that relate attention reconstruction error to clustering quality, and empirically show that `SVG-EAR` improves the quality-efficiency trade-off and increases throughput at the same generation fidelity on video diffusion tasks. Overall, `SVG-EAR` establishes a clear Pareto frontier over prior approaches, achieving up to 1.77× and 1.93× speedups while maintaining PSNRs of up to 29.759 and 31.043 on Wan2.2 and HunyuanVideo, respectively.

**Keywords:** Video Generation · Sparse Attention · Video Diffusion Model

## 1 Introduction

Diffusion transformers (DiTs) have become a dominant paradigm for high-fidelity image and video generation [25, 45]. However, in video generation settings, the token sequence length grows rapidly with resolution and frame count, while the quadratic cost of attention quickly becomes a primary bottleneck. As a result,

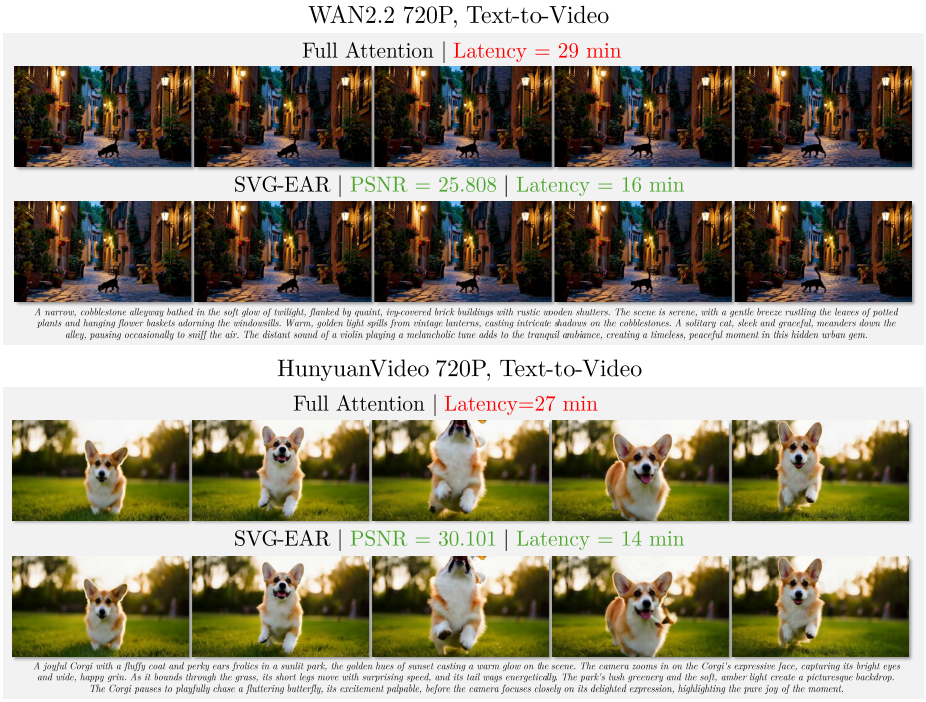



**Fig. 1:** `SVG-EAR` significantly accelerates video generation for Wan2.2 and HunyuanVideo. On a single NVIDIA H100 GPU, achieving 1.81× and 1.93× speedups with 26 and 30 PSNR, respectively.

accelerating attention *without* noticeable quality degradation is increasingly critical for long-form and high-resolution video diffusion [51, 54, 67].

A large body of work tackles this bottleneck via sparse attention. A particularly practical family exploits *structured redundancy* in attention maps: tokens are semantically clustered and permuted so that similar tokens become contiguous in memory, turning the attention matrix into a block structure over query clusters and key clusters. Then, only a subset of query–key blocks are computed exactly (block-sparse), while the rest are ignored or approximated, yielding hardware-friendly sparsity patterns and substantial speedups [54]. This "cluster–permute–route" pipeline is appealing in video DiTs because it aligns well with efficient kernels and preserves important structure.

Despite these advances, existing methods face a fundamental tension in block selection and the treatment of unselected blocks. Many approaches select blocks by approximated attention scores (*e.g.,* top-$k$/top-$p$) and *drop* low-score blocks entirely. While intuitive, low-score blocks can still collectively carry important global context (*e.g.,* background consistency, long-range semantic coupling, and weak but numerous dependencies). Consequently, naively discarding low-score blocks incurs non-trivial information loss and can manifest as perceptible quality

degradation in video generation. Recent works such as SLA [63] and SLA2 [64] address this by pairing sparse exact attention with a learned linear branch that approximates the contributions of dropped blocks, recovering much of the lost context. However, this approach introduces additional trainable parameters and requires fine-tuning, limiting its plug-and-play applicability and sacrificing the fidelity.

A natural solution is to avoid pure dropping by leveraging the high within-block similarity revealed by clustering. For blocks that are not selected for exact computation, one can apply a *parameter-free linear compensation* using centroids: replace queries and keys within a cluster with their centroid and approximate the contribution of an entire block with a shared interaction. This branch requires no training and no additional parameters, and it mitigates information loss from dropped blocks.

However, linear compensation alone does not resolve the core issue. Crucially, once a compensation branch exists, conventional score-based routing becomes *misaligned* with the objective of controlling the final approximation error. A high-score block may be highly coherent within the cluster and thus well approximated by its centroid, making exact computation unnecessary. Conversely, a low-score block can contain diverse key-value interactions where centroid-based compensation induces substantial error. Therefore, under a fixed compute budget, the goal should not be to preserve the highest-score blocks, but to *minimize the reconstruction error* between full attention and its compensated counterpart, allocating exact computation to the blocks where compensation fails.

Motivated by this insight, we propose `SVG-EAR`, a sparse attention method that combines *parameter-free linear compensation* with *error-aware routing*. `SVG-EAR` first clusters queries and keys to form a block structure based on similarity. It then computes exact attention for a subset of blocks under the density budget assigned by users, and approximates the remaining blocks using linear compensation with key/value cluster centroids. Unlike prior score-based selection, `SVG-EAR` estimates the *compensation error* of each block using a lightweight probing procedure and greedily selects blocks with the highest *error-to-cost ratio* (*i.e.,* the predicted compensation error normalized by the block size) under a density budget. To make routing practical at inference time, we exploit intra-cluster similarity and use query centroids as proxies for individual queries, reducing the estimation cost from quadratic $\mathcal{O}(N_q N_k d)$ to near-linear $\mathcal{O}(C_q N_k d)$. We further implement a fused, streaming kernel to avoid materializing intermediate logits and to keep the routing overhead negligible in practice.

We validate the proposed design both theoretically and empirically. On the theory side, we provide an upper bound that relates the true attention-map error to our estimated error, characterizing its dependence on clustering quality and sequence length. On the empirical side, `SVG-EAR` achieves a superior error–density trade-off on video generation: at the same density, it reduces attention reconstruction error and improves generation quality, and at the same quality target it operates at lower density and delivers higher throughput (see §5). Specifically, `SVG-EAR` establishes a clear Pareto frontier over prior approaches [51, 54, 67],

reaching up to 1.77× and 1.93× speedups while maintaining PSNRs of up to 29.759 (Wan2.2 [45]) and 31.043 (HunyuanVideo [25]), respectively. Overall, our results suggest that when compensation is available, the key to high-fidelity sparse attention is not "*selecting high-score blocks*", but identifying where compensation breaks and prioritizing those blocks for exact computation.

We summarize our key contributions as follows:

1. We identify two fundamental misalignments in score-based sparse attention: (i) naively *dropping* low-attention-score blocks can cause substantial information loss; and (ii) once an approximation/compensation branch is introduced, block selection should *not* prioritize high-score blocks, but instead prioritize blocks that would otherwise incur *large approximation error*.
2. We prototype and implement a compensation-and-routing sparse attention mechanism: a parameter-free *linear compensation* branch that recovers contributions from uncomputed blocks via cluster means, and an *error-aware routing* strategy that, under a fixed budget, identifies and computes the blocks with the largest induced error, yielding a markedly improved error–density trade-off.
3. We further translate the design into an end-to-end system with efficient kernels and execution flow that keep the overhead negligible in practice, delivering substantial and consistent speedups on real video generation workloads while maintaining generation fidelity.

## 2 Related Work

**Sparse Attention for Video Generation.** Sparse attention is a leading direction for accelerating video Diffusion Transformers [62] because 3D spatiotemporal self-attention scales quadratically with the huge token sequence length. Sparse attention mechanisms for video generation can be roughly divided into static and dynamic approaches, based on whether the sparsity pattern is fixed offline or determined online at inference time. Static approaches typically exploit recurring structural patterns [7,23,30,51,58,70]. Dynamic approaches infer masks or critical-token sets per sample and timestep, commonly introducing an identification/proxy-scoring step whose overhead is usually designed to be negligible [4, 5, 26, 29, 36, 43, 47, 52, 54, 67]. Beyond training-free methods, newer "trainable sparse attention" lines push sparsity higher with fine-tuning and additional branch compensation [1, 32, 49, 61, 63, 64, 69].

**Linear Attention for Diffusion Models.** Linear attention and state-space alternatives are important for video diffusion because quadratic attention quickly dominates latency and memory as spatiotemporal token counts grow. Representative families can be grouped into kernelized linear attention [6], state-space/SSM-centered designs that use structured recurrence for global mixing [16, 17, 46], and hybrid local–global designs that explicitly balance a cheap linear surrogate with selective higher-fidelity pathways [13, 63]. Though these methods enable predictable memory scaling and improved feasibility of long-context

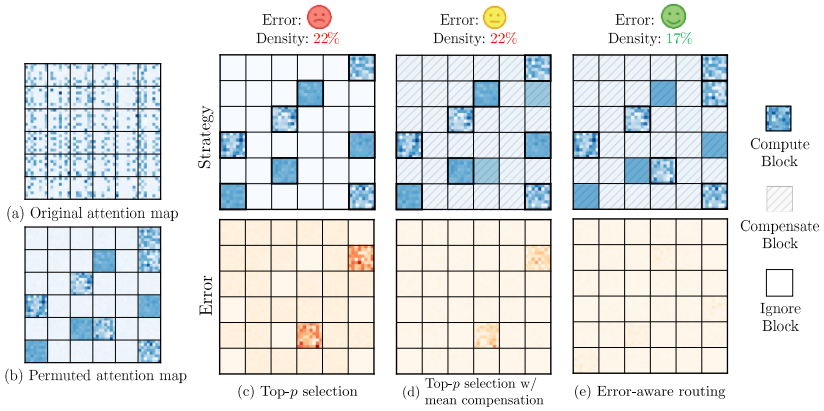



**Fig. 2:** Existing methods and top-$p$ selection fall short: *dropping "low-score" blocks* and *error-unaware block selection* degrade the error–density trade-off. (a) Original attention map. (b) Permuted map after semantic-aware clustering. (c) Ignoring low-score blocks causes a large, sparse attention error. (d) Linear compensation with cluster means still yields high error due to naive top-$p$ selection. (e) Our method improves both error and density by routing based on the gap between full computation and compensation.

generation under fixed budgets, they still suffer from low quality and long-term dependencies.

**Other Optimization Techniques.** Cache-based acceleration exploits redundancy across denoising steps or between conditional/unconditional branches in classifier-free guidance [3,9,24,33,39–42]. Parallelization becomes essential when targeting deployment-scale throughput or higher resolutions [10–12, 14, 22, 27, 34]. Quantization reduces memory bandwidth and leverages low-precision tensor cores to accelerate further the linear module [28, 31, 48] or attention module [59, 60, 65, 66, 68]. Distillation and few-step sampling compress long diffusion trajectories into a small number of steps, usually leading to more than 10× speedup [37, 55, 56, 71].

**Autoregressive and Streaming Video Generation.** Autoregressive (streaming) video diffusion generates frames (or chunks) sequentially and enables efficient KV caching, usually accompanied by unbounded temporal horizons and interactive control. CausVid [57] and Self-Forcing [18] exemplify this shift by converting bidirectional video diffusion into causal generation that better matches interactive inference and is consolidated through further algorithmic innovations to mitigate anti-drifting [8,35,38,53,72]. World Models [2,15,20,21,44,50] objectives overlap with streaming video generation but extend it toward closed-loop simulation, where an agent or user can intervene at each step.

# 3 Motivation

## 3.1 Structured Redundancy in Attention Computation

Previous works exploit the inherent sparsity in attentions to accelerate DiTs [51, 54,67,70]. We first cluster semantically similar tokens and permute them so that tokens within each cluster are laid out contiguously. This allows the attention map to be partitioned into blocks that often exhibit high inner-block similarity. As shown in Fig. 2 (c), the existing methods then select a subset of these blocks for exact attention ranked by their approximated attention scores, while simply ignoring the low score blocks. However, this approach can lead to significant information loss, as the ignored blocks can also contain important contextual information, and the limited budget may not allow for their inclusion.

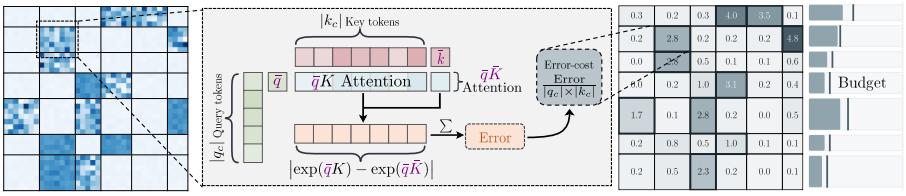
Fortunately, the similarities within each block naturally provide an opportunity to efficiently approximate the skipped blocks' contributions without any training and additional parameters. As shown in Fig. 2 (d), a simple strategy here is to use the mean of the remaining blocks in each cluster to fill in the gaps left by the ignored blocks. Given a block to compensate, suppose the mean of the key-value tokens is $\bar{k}$. The attention logits for all entries in the block interacted with a query $q_i$ can be then computed as $q_i\bar{k}^T/\sqrt{d}$, where $d$ is the dimensionality of the token vectors.

## 3.2 Existing Block Selection Fails to Control Output Error

While linear compensation improves the error–density trade-off, its effectiveness ultimately depends on how the blocks are selected. Existing block selection strategies are score-based and thus misaligned with the objective of controlling the final output error when linear compensation is introduced. For example, a block with high attention scores may exhibit strong inner-block similarity, making it well suited for linear compensation. In contrast, a block with low attention scores can contain diverse key–value interactions, where linear compensation induces substantial approximation error. To maximize fidelity under a fixed compute budget, the goal should be to minimize the reconstruction error of the attention map. As shown in Fig. 2 (e), our proposed approach routes computation based on the gap between the full attention map and its linearly compensated counterpart. As a result, by directly optimizing an error-aware objective and prioritizing the blocks that contribute most to the reconstruction error, we achieve a superior error–density trade-off, as detailed below.

# 4 Methodology

In this section, we first formulate the problem based on our parameter-free linear compensation. We then introduce `SVG-EAR`, an error-aware routing method with theoretical guarantees and low overhead. Our key insight is that the error between the full attention map and its linearly compensated counterpart can




(a) Permuted attention map (b) Block error estimation (c) Error-to-size ratio map

**Fig. 3:** Overview of `SVG-EAR` (a) The attention map after semantic-aware clustering. (b) Block error estimation. Using the cluster mean as a proxy for individual queries, the block error is computed via the sum of squared differences between exponentiated logits of individual keys $k_j$ and the key mean $\bar{k}$, then normalized by the block area to determine the error-to-size ratio. (c) Blocks with the highest error-to-size ratio are greedily selected for exact attention within the budget, while the rest are assigned to linear compensation.

be accurately estimated via a lightweight probing algorithm, which enables precise attention routing. We further show how the linear compensation leverages value-cluster means to produce the final output efficiently. Finally, we implement a fused kernel that estimates the error using a streaming update.

### 4.1 Problem Formulation

Given a transformer attention layer with query tokens $Q \in \mathbb{R}^{N_q \times d}$ and key tokens $K \in \mathbb{R}^{N_k \times d}$. We cluster $Q$ and $K$ using flash k-means [54], and obtain $C_q$ and $C_k$ clusters, respectively. For each query $q_i$ and key $k_j$, we use $\bar{q}_i$ and $\bar{k}_j$ to denote the mean token in the cluster that contains $q_i$ and $k_j$, respectively. We denote $\bar{Q} \in \mathbb{R}^{N_q \times d}$ and $\bar{K} \in \mathbb{R}^{N_k \times d}$ as the matrices of per-token cluster means for $Q$ and $K$, respectively; that is, the $i$-th row of $\bar{Q}$ equals $\bar{q}_i$ (the centroid of the cluster containing $q_i$), and similarly the $j$-th row of $\bar{K}$ equals $\bar{k}_j$.

Our core idea here is to approximate part of the attention computation linearly with cluster means. For a query $q_i$, when approximating its interaction with a cluster of keys $k_c = \{k_j\}$, we replace each key $k_j$ in the logit computation with the corresponding cluster mean $\bar{k}_j$. This gives an approximate logit $q_i \bar{k}_j^T / \sqrt{d}$, which is shared across all keys within the same cluster.

We use a binary routing mask $M \in \{0,1\}^{N_q \times N_k}$ for each query $q_i$. If $M_j^{(i)} = 1$, we compute the interaction between $q_i$ and $k_j$ exactly. If $M_j^{(i)} = 0$, we approximate it using the cluster mean. To match ML accelerator-friendly sparsity, where the tokens in the same cluster will be permutated continuously and processed together, the mask should be applied on the query-key block level: All queries in the same cluster share the same mask. For a fixed query, all keys in the same cluster share the same mask value. With this mask, we write the sparse attention map as

$$\tilde{A}^{(M)}_{\text{sparse}} = \text{softmax}\left(\frac{QK^T}{\sqrt{d}} \odot M + \frac{Q\bar{K}^T}{\sqrt{d}} \odot \big(1 - M\big)\right) \tag{1}$$

Given a budget $\rho$ (*i.e.,* density), our goal is to find a mask $M$ that minimizes the mean squared error (MSE) between the full attention map and the sparse attention map, subject to the constraint that the number of non-zero entries in $M$ is at most $\rho N_q N_k$. Formally, we formulate the optimization problem as:

$$\min_{M} \|\tilde{A}^{(M)}_{\text{sparse}} - A_{\text{full}}\|_F^2 \quad \text{s.t.} \quad \|M\|_F^2 \leq \rho N_q N_k, \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $A_{\text{full}} = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)$ is the full attention map.

### 4.2 Error-Aware Routing

**Relaxed optimization problem.** The above optimization problem is difficult to solve efficiently at routing time because the softmax normalizer couples all keys for each query, and $A_{\text{full}}$ and $\tilde{A}^{(M)}_{\text{sparse}}$ use different normalization terms. In particular, for any query, altering the routing decision for any single key cluster changes the normalizer and thus changes the attention weights assigned to all other keys. Consequently, selecting an optimal block-level mask cannot be reduced to independent per-block choices; in the worst case it requires searching over a combinatorial space of size up to $\mathcal{O}(2^{C_q \cdot C_k})$, which is intractable.

To obtain a practical and block-separable proxy, we therefore relax the objective by focusing on the exponentiated logits as follows:

$$\min_{M} \sum_{i,j} \left(1 - M_j^{(i)}\right) \left( \exp(\frac{q_i \bar{k}_j^T}{\sqrt{d}}) - \exp(\frac{q_i k_j^T}{\sqrt{d}}) \right)^2 \quad \text{s.t.} \quad \|M\|_F^2 \leq \rho N_q N_k. \tag{3}$$

Intuitively, this relaxation shifts the objective from finding a mask $M$ that minimizes the discrepancy between two normalized attention maps to minimizing the discrepancy between their unnormalized attention weights. Although this relaxation may fail to identify masks whose near-optimality only emerges after normalization, it still provides a useful proxy for routing decisions. This is because, in our setting, the variation in the softmax normalizer is practically limited for two reasons: (1) semantic clustering ensures that each $k_j$ is close to its centroid $\bar{k}_j$, reducing deviation in the logits; (2) the unmasked tokens typically receive high attention scores, so they dominate the normalization term, thereby limiting the impact of the remaining masked tokens on the normalizer.

**Oracle solution.** For the relaxed optimization problem, an oracle solution can in principle be obtained by selecting the query–key blocks that incur the largest squared errors under the budget constraint, which reduces to a tractable 0–1 knapsack problem. Specifically, let

$$\epsilon_{i,j}^2 = \left( \exp\left(\frac{q_i \bar{k}_j^T}{\sqrt{d}}\right) - \exp\left(\frac{q_i k_j^T}{\sqrt{d}}\right) \right)^2 \tag{4}$$

denote the squared error for the entry $(i, j)$. Each query-key block $(q_c, k_c)$ can then be treated as an item in the knapsack formulation, where the value is given

by $\sum_{(i,j)\in(q_c,k_c)} \epsilon_{i,j}^2$, and the weight corresponds to its size $|q_c||k_c|$. After that, the problem can be solved using standard knapsack algorithms or well approximated by greedy methods that select blocks in descending order of error-to-cost ratio $\frac{\sum_{(i,j)\in(q_c,k_c)} \epsilon_{i,j}^2}{|q_c||k_c|}$ until the budget is exhausted.

The oracle solution is fully error-aware; however, it is not practical. Computing the exact per-entry error requires evaluating the full attention logits for all token pairs, which incurs the same $\mathcal{O}(N_q N_k d)$ quadratic cost as full attention itself. In other words, the oracle depends on precisely the information that sparse attention is designed to avoid computing.

**Error estimation.** To achieve error-aware routing without incurring the prohibitive cost of computing the exact error, we propose to estimate the error using a lightweight probing algorithm. Again, we leverage the inherent similarity within each query token cluster, which allows us to use the cluster mean as a proxy for the individual queries in the error estimation. Specifically, we define the following estimated error for each entry $(i, j)$:

$$\hat{\epsilon}_{i,j}^2 = \big(\exp\big(\frac{\bar{q}_i \bar{k}_j^T}{\sqrt{d}}\big) - \exp\big(\frac{\bar{q}_i k_j^T}{\sqrt{d}}\big)\big)^2 \tag{5}$$

Because queries within the same cluster share a common estimated error, the computational complexity reduces to $\mathcal{O}(C_q N_k d)$, which is negligible in practice when an optimized kernel implementation is used.

We then use the estimated errors to determine whether each query–key block is assigned to exact attention or linear compensation. Specifically, we first aggregate the estimated errors within each block. We then greedily select the blocks with the highest error-to-size ratio $\sum_{(i,j)\in(q_c,k_c)} \hat{\epsilon}_{i,j}^2 / |q_c||k_c|$ for exact attention until the budget is exhausted, while assigning the remaining blocks to linear compensation.

**Proposition 4.1.** *Let $\delta_q^2 = \frac{1}{N_q}\sum_i \|q_i - \bar{q}_i\|_2^2$ denote the average squared $\ell_2$ error between each query and its cluster mean, and let $K_{\max}$ denote the maximum $\ell_2$ norm of the key tokens. Given any mask $M$ that does not significantly perturb the attention normalizers, we have:*

$$\frac{1}{N_q N_k}\|\tilde{A}_{sparse}^{(M)} - A_{full}\|_F^2 \;\le\; \frac{2}{N_q N_k}\sum_{i,j}(1 - M_j^{(i)})\,\frac{\hat{\epsilon}_{i,j}^2}{Z_i^2} \;+\; \frac{8\delta_q^2 K_{\max}^2}{N_k d} \tag{6}$$

*, where $Z_i$ denotes the softmax normalizer for query $i$.*

See Supplementary Material for the proof and the formal statement of the normalizer stability assumption. The left-hand side is the true MSE on the attention map, while the right-hand side consists of the estimated MSE (using $\hat{\epsilon}_{i,j}^2$) scaled by a constant factor, plus a residual term. The residual $\frac{8\delta_q^2 K_{\max}^2}{N_k d}$ scales linearly with the clustering error $\delta_q^2$ and decreases inversely with the sequence length $N_k$. Therefore, as clustering improves (*i.e.,* $\delta_q^2 \to 0$) or the sequence length increases, the bound becomes asymptotically tight, establishing that our error estimation is both theoretically safe and quantitatively controllable.

### 4.3 Linear Compensation

After determining the mask $M$, we compute the final attention output by combining exact and approximated contributions. Let $V \in \mathbb{R}^{N_k \times d}$ denote the value tokens, which follow the same clustering assignment as $K$, and let $\bar{V} \in \mathbb{R}^{N_k \times d}$ denote the matrix of mean value tokens, where each $\bar{v}_j$ is the average of $V$ over the key cluster containing $v_j$. The sparse attention output is then given by:

$$\tilde{O}^{(M)}_{\text{sparse}} = \big(\tilde{A}^{(M)}_{\text{sparse}} \odot M\big)V + \big(\tilde{A}^{(M)}_{\text{sparse}} \odot (1 - M)\big)\bar{V} \tag{7}$$

For blocks where $M^{(i)}_j = 1$, the output is computed with the original values $V$. For blocks where $M^{(i)}_j = 0$, since all keys within the cluster share the same approximate attention weight, the contribution of a key cluster $k_c$ reduces to $|k_c| \cdot \tilde{A}^{(M)}_{\text{sparse}}(i,j) \cdot \bar{v}_j$. This significantly improves computational efficiency: for a given query and masked block, we only need to compute a single logit using the cluster mean $\bar{k}_j$ and a single scalar–vector product with the mean value $\bar{v}_j$. Hence, the overall time complexity of `SVG-EAR`'s attention computation is reduced to $\mathcal{O}\Big((N_q C_k + \rho N_k N_q) \cdot d\Big)$.

Moreover, since the linear compensation also considers the value tokens, we can further improve the error estimation above by incorporating each $v_j$ and $\bar{v}_j$ into the analysis. A *value-aware error estimation* is thus given by:

$$\tilde{\epsilon}^2_{i,j} = \| \exp\big(\frac{\bar{q}_i \bar{k}_j^T}{\sqrt{d}}\big)\bar{v}_j - \exp\big(\frac{\bar{q}_i k_j^T}{\sqrt{d}}\big) v_j \|_2^2 \tag{8}$$

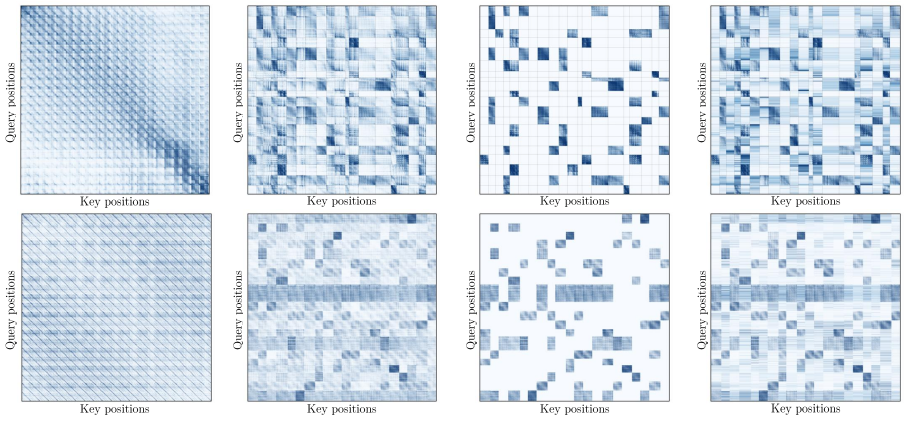
which can be computed in the same complexity $\mathcal{O}(C_q N_k d)$ as Equation 5 while providing a practically more accurate estimation.

### 4.4 Efficient Kernel Implementation

The error estimation procedure is inherently memory I/O–bound, as naively materializing all logits requires repeatedly writing intermediate results to HBM. To address this bottleneck, we design a custom kernel based on a streaming update scheme. Specifically, instead of materializing all logits in HBM and computing the squared errors afterward, we fuse these steps by expanding the squared terms and maintaining the required running statistics on the fly. To ensure numerical stability, we subtract the running maximum from the logits prior to exponentiation. This design reduces HBM accesses to $\mathcal{O}(N_k C_q d \cdot \mathbf{S}^{-1})$, where $\mathbf{S}$ denotes the SRAM size, while incurring only a small constant-factor increase in FLOPs within the same $\mathcal{O}(N_k C_q d)$ complexity. For the linear compensation part, we reuse SVG2's [54] customized attention kernel that accepts dynamic block sizes as inputs. See Supplementary Material for more implementation details.

## 5 Experiments

**Models.** We evaluate `SVG-EAR` on open-sourced state-of-the-art video generation models, including Wan2.2-I2V/T2V-A14B [45], and HunyuanVideo-T2V-




(a) Original attention map (b) Permuted attention map (c) Top-p selection (SVG2) (d) Error-aware routing w/ mean compensation

**Fig. 4:** Visualization of attention maps from specific attention heads in Wan2.2 during text-to-video generation. (a) Original attention maps with sparse patterns, using heatmaps to represent attention weights. (b) Permuted attention maps. Following SVG2, we permute the attention maps such that attention begins to cluster into specific blocks. (c) Applying SVG2's top-$p$ selection to these permuted attention maps, which ignores the sparse attention mechanisms of the remaining blocks. (d) Applying our Error-aware Routing and Mean Compensation mechanisms to the permuted attention maps, achieving higher attention map similarity.

13B [25] to generate videos with 720p resolution. After being tokenized by 3D-VAE, Wan2.2 generates 21 frames with 3,600 tokens per frame, while Hunyuan-Video processes 33 frames with 3,600 tokens per frame.

**Metrics.** We assess the similarity of the generated video compared to full attention using the following metrics: Peak Signal-to-Noise Ratio (PSNR), Learned Perceptual Image Patch Similarity (LPIPS), and Structural Similarity Index Measure (SSIM). We use VBench [19] to evaluate the video quality. To quantify the efficiency of sparse attention mechanisms (*i.e.,* computational budget), we use density, which is defined as the sparse attention computation divided by the full attention computation. To assess end-to-end efficiency, we test the Speedup Ratio, which is calculated as the inference time of the full attention divided by the inference time of the sparse attention.

**Datasets.** For text-to-video generation, we adopt the prompt in Penguin Benchmark after prompt optimization provided by the VBench team. For image-to-video generation, we adopt the prompt-image pairs provided by VBench with prompt enhancement. We evaluate the video quality on a subset of 50 samples randomly selected from text-to-video and image-to-video datasets, respectively.

**Baselines.** We compare `SVG-EAR` against state-of-the-art sparse attention algorithms, including static method Sparse VideoGen (SVG) [51], and dynamic methods Sparse VideoGen 2 (SVG2) [54] and SpargeAttention [67]. We do not include SLA [63, 64] because it requires extra training, which makes similarity

**Table 1:** Quality and efficiency benchmarking results of `SVG-EAR` and baselines, where the best results are highlighted, and the second-best results are underlined.

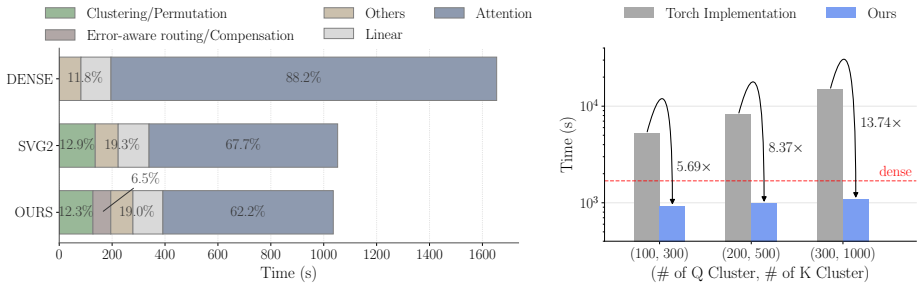
| Config | PSNR↑ | SSIM↑ | LPIPS↓ | ImgQual↑ | SubCons↑ | Density↓ | FLOP↓ | Speedup↑ |
|---|---|---|---|---|---|---|---|---|
| *Wan 2.2 14B, 720P, I2V* | - | - | - | 0.704 | 0.960 | 100% | 658.46 PFLOPS | 1× |
| SpargeAttn | 27.140 | 0.883 | 0.116 | 0.703 | 0.958 | 30.15% | 396.83 PFLOPS | 1.58× |
| SVG | 25.297 | 0.844 | 0.139 | 0.703 | 0.958 | 30.25% | 397.20 PFLOPS | 1.58× |
| SVG2 | 27.668 | 0.888 | 0.117 | 0.701 | 0.958 | 29.38% | 393.95 PFLOPS | 1.61× |
| `SVG-EAR` | **29.759** | **0.918** | **0.093** | **0.704** | **0.959** | 23.64% | 378.88 PFLOPS | 1.61× |
| `SVG-EAR`-Turbo | 28.344 | 0.900 | 0.108 | 0.702 | 0.958 | **20.42%** | 363.85 PFLOPS | **1.77**× |
| *Wan 2.2 14B, 720P, T2V* | - | - | - | 0.706 | 0.916 | 100% | 658.46 PFLOPS | 1× |
| SpargeAttn | 20.872 | 0.708 | 0.242 | 0.708 | **0.916** | 30.15% | 396.83 PFLOPS | 1.58× |
| SVG | 19.455 | 0.654 | 0.292 | **0.712** | 0.912 | 30.25% | 397.20 PFLOPS | 1.59× |
| SVG2 | 23.556 | 0.802 | 0.183 | 0.705 | 0.914 | 32.30% | 404.88 PFLOPS | 1.57× |
| `SVG-EAR` | **24.995** | **0.841** | **0.153** | 0.706 | 0.915 | 25.95% | 387.53 PFLOPS | 1.59× |
| `SVG-EAR`-Turbo | 23.940 | 0.814 | 0.174 | 0.705 | 0.915 | **22.25%** | 370.71 PFLOPS | **1.75**× |
| *Hunyuan 13B, 720P, T2V* | - | - | - | 0.665 | 0.904 | 100% | 612.38 PFLOPS | 1× |
| SpargeAttn | 24.589 | 0.796 | 0.232 | 0.629 | **0.908** | 40.09% | 389.76 PFLOPS | 1.38× |
| SVG | 27.325 | 0.880 | 0.140 | **0.665** | 0.905 | 29.92% | 351.97 PFLOPS | 1.57× |
| SVG2 | 29.445 | 0.911 | 0.112 | 0.654 | 0.901 | 26.21% | 299.02 PFLOPS | 1.89× |
| `SVG-EAR` | **31.043** | **0.928** | **0.092** | 0.659 | 0.903 | 22.17% | 281.86 PFLOPS | **1.93**× |

metrics not comparable. We provide the details of each method in Supplementary Material.

**Implementation.** We implement `SVG-EAR` on top of SVG2 [54], inheriting its budget allocation strategy and default hyperparameters. Since SVG2 assigns an adaptive compute budget to each query cluster, our routing algorithm operates at the granularity of individual query clusters accordingly. By reducing the number of centroids and the density budgets, we developed `SVG-EAR`-turbo that matches the efficiency of SVG2-turbo while consistently outperforming SVG2 in quality.

### 5.1 Quality Evaluation

We present the attention maps of specific heads during the generation process of Wan2.2 Text-to-Video on prompts from VBench, exhibiting various sparse patterns as shown in Fig. 4. Compared to the original top-p selection, our method covers a broader range and allocates attention to marginal tokens with relatively small attention weights. The attention maps generated by our method are more similar to the permuted maps, particularly in cases where the attention weight distribution is relatively uniform. In addition, we empirically validate the normalizer stability assumption by replacing dense attention with `SVG-EAR`, which leads to only a 4.18% average relative change in the softmax normalizer.




**(a)** Generation latency breakdown **(b)** Efficiency of Custom Kernels

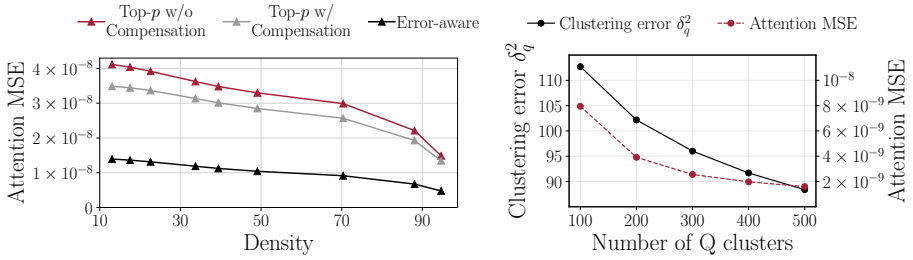
**Fig. 5:** Efficiency evaluation. (a) illustrates the latency breakdown of different components during a single complete inference process of Wan2.2 T2V 720p, compared with the vanilla implementation and SVG2. (b) presents an end-to-end latency comparison between our efficient Triton implementation and the native PyTorch version.

The quantitative results summarized in Table 1 demonstrate that `SVG-EAR` consistently outperforms all baselines across PSNR, LPIPS, and SSIM, while simultaneously achieving the highest speedup. Specifically, `SVG-EAR` achieves an average PSNR of **24.995** on Wan2.2 T2V and **31.043** on HunyuanVideo. Furthermore, compared to the existing SOTA SVG2 [54], `SVG-EAR` establishes a new Pareto frontier, representing a superior trade-off between generation quality and inference efficiency. For a more comprehensive evaluation, see Supplementary Material. Moreover, for reference-free metrics, we present representative VBench metrics in Table 1, while the comprehensive evaluation is detailed in Supplementary Material.

### 5.2 Efficiency Evaluation

**End-to-end speedup evaluation.** We evaluate the efficiency of `SVG-EAR` by measuring the end-to-end inference time speedup compared to full attention and total FLOPS, as shown in Table 1. `SVG-EAR` achieves a speedup of **1.59**× on Wan2.2 T2V and **1.93**× on HunyuanVideo, higher than all baselines. Notably, `SVG-EAR`-turbo achieves a speedup of **1.75**× on Wan2.2 T2V, while maintaining superior quality compared to all baselines.

**Kernel efficiency evaluation.** We evaluated the computational overhead of our method. As shown in Fig. 5 (a), our approach accounts for 6.5% of the total end-to-end latency during Wan2.2 text-to-video (720p) inference. While the latency reduction is minor, the results are noticeably better in visual fidelity and overall quality. The efficiency is driven by our custom Triton kernel, which achieves up to a 13.74× speedup compared to the original PyTorch implementation as shown in Fig. 5 (b). This allows our method to enhance generation quality while maintaining a high inference speed.




**(a)** Attention MSE vs. Density **(b)** Clustering Error

**Fig. 6:** Error Analysis. (a) Comparison of attention MSE versus density (*i.e.,* number of sparse computation normalized by total computation) for top-$p$ selection, top-$p$ selection with linear compensation, and error-aware routing with linear compensation. (b) Both the clustering error $\delta_q^2$ (solid black line) and attention MSE (dashed red line) decrease as the number of Q clusters increases, demonstrating that higher clustering quality leads to a tighter error bound and better approximation.

### 5.3 Error Analysis

**Strategy of block selection.** To access the error-density trade-off of the three methods illustrated in Fig. 2, *i.e.,* top-$p$ selection (SVG2), top-$p$ selection with linear compensation, and error-aware routing with linear compensation, we compute the mean squared error between each method's attention map and the full attention map, *i.e.,* $\frac{1}{N_q \times N_k}\|A_{\text{sparse}} - A_{\text{full}}\|_F^2$. Following the primary experimental settings with $Q_C = 300$ and $K_C = 1,000$, we conduct this evaluation by sampling a representative set of attention maps across the generation process. As shown in Fig. 6 (a), our error-aware routing yields the attention map most consistent with the full attention, while the top-$p$-based compensation only achieves a marginal error reduction against SVG2.

**Effect of clustering.** We provide a theoretical guarantee that relates attention reconstruction error to clustering quality of query tokens in §4. To evaluate the effect, we conduct inference on an additional sample with $K_C = 1,000$ and $p = 0.85$. As shown in Fig. 6 (b), we compute the clustering error $\delta_q^2$ and the attention MSE across varying numbers of query centroids (*i.e.,* $Q_C$). We observe a significant reduction in attention error as $Q_C$ increases, demonstrating that improved clustering quality directly enhances the approximation accuracy. Notably, when $Q_C$ is set to 300, the relative variation in reconstruction error remains within 4.75%, suggesting that the approximation remains stable under this clustering configuration. This empirical observation aligns with our theoretical guarantees, where the attention error relates to the clustering quality.

## 6 Conclusion and Limitation

We presented `SVG-EAR`, a training-free approach to accelerate DiT-based video generation under sparse attention. `SVG-EAR` recovers the contribution of skipped

attention blocks via a parameter-free centroid compensation branch, leveraging the strong similarity structure of keys and values after semantic clustering. To avoid the small fraction of blocks where compensation is inaccurate, we further introduced error-aware routing that prioritizes blocks with the highest error-to-cost ratio under a fixed density budget. We provided theoretical guarantees linking attention reconstruction error to clustering quality, and demonstrated empirically that `SVG-EAR` improves the quality–efficiency trade-off. The main limitation of this paper is the lack of discussion and evaluation of whether the proposed method extends to attention mechanisms beyond DiTs.

## Acknowledgements

This work is supported by NSF awards IIS-1955488, IIS-2027575, DOE awards DE-SC0016260, AC02-05CH11231, and DARPA Agreement No. HR00112590131.

## References

1. Agarwal, K., Chen, Z., Luo, C., Chen, Y., Zheng, H., Huang, X., Rudra, A., Chen, B.: Monarchrt: Efficient attention for real-time video generation. arXiv preprint arXiv:2602.12271 (2026)
2. Bruce, J., Dennis, M.D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., et al.: Genie: Generative interactive environments. In: Forty-first International Conference on Machine Learning (2024)
3. Bu, J., Ling, P., Zhou, Y., Wang, Y., Zang, Y., Lin, D., Wang, J.: Dicache: Let diffusion model determine its own cache. arXiv preprint arXiv:2508.17356 (2025)
4. Cai, S., Yang, C., Zhang, L., Guo, Y., Xiao, J., Yang, Z., Xu, Y., Yang, Z., Yuille, A., Guibas, L., et al.: Mixture of contexts for long video generation. arXiv preprint arXiv:2508.21058 (2025)
5. Chen, A., Liu, Y., Huang, J., Lian, G., Yao, Y., Lan, W., Lin, J., Ma, Z., Zhou, T., Yang, H.: Rainfusion2. 0: Temporal-spatial awareness and hardware-efficient block-wise sparse attention. arXiv preprint arXiv:2512.24086 (2025)
6. Chen, J., Zhao, Y., Yu, J., Chu, R., Chen, J., Yang, S., Wang, X., Pan, Y., Zhou, D., Ling, H., et al.: Sana-video: Efficient video generation with block linear diffusion transformer. arXiv preprint arXiv:2509.24695 (2025)
7. Chen, R., Mills, K., Jiang, L., Gao, C., Niu, D.: Re-ttention: Ultra sparse visual generation via attention statistical reshape. In: Belgrave, D., Zhang, C., Lin, H., Pascanu, R., Koniusz, P., Ghassemi, M., Chen, N. (eds.) Advances in Neural Information Processing Systems. vol. 38, pp. 58029–58055. Curran Associates, Inc. (2025)
8. Chen, S., Wei, C., Sun, S., Nie, P., Zhou, K., Zhang, G., Yang, M.H., Chen, W.: Context forcing: Consistent autoregressive video generation with long context. arXiv preprint arXiv:2602.06028 (2026)
9. Chu, H., Wu, W., Feng, G., Zhang, Y.: Omnicache: A trajectory-oriented global perspective on training-free cache reuse for diffusion transformer models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16302–16312 (2025)

10. Chu, X., Li, R., Wang, Y.: Usp: Unified self-supervised pretraining for image generation and understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18475–18486 (2025)
11. Fang, J., Pan, J., Li, A., Sun, X., Wang, J.: Pipefusion: Patch-level pipeline parallelism for diffusion transformers inference. arXiv preprint arXiv:2405.14430 (2024)
12. Fang, J., Pan, J., Sun, X., Li, A., Wang, J.: xdit: an inference engine for diffusion transformers (dits) with massive parallelism. arXiv preprint arXiv:2411.01738 (2024)
13. Fang, T., Zhang, H., Xie, R., Han, Z., Tao, X., Zhao, T., Wan, P., Ding, W., Ouyang, W., Ning, X., et al.: Salad: Achieve high-sparsity attention via efficient linear attention tuning for video diffusion transformer. arXiv preprint arXiv:2601.16515 (2026)
14. Feng, T., Li, Z., Yang, S., Xi, H., Li, M., Li, X., Zhang, L., Yang, K., Peng, K., Han, S., et al.: Streamdiffusionv2: A streaming system for dynamic and interactive video generation. arXiv preprint arXiv:2511.07399 (2025)
15. Gao, J., Chen, Z., Liu, X., Zhuang, J., Xu, C., Feng, J., Qiao, Y., Fu, Y., Si, C., Liu, Z.: Longvie 2: Multimodal controllable ultra-long video world model. arXiv preprint arXiv:2512.13604 (2025)
16. Gao, Y., Huang, J., Sun, X., Jie, Z., Zhong, Y., Ma, L.: Matten: Video generation with mamba-attention. arXiv preprint arXiv:2405.03025 (2024)
17. Huang, J., Zhang, G., Jie, Z., Jiao, S., Qian, Y., Chen, L., Wei, Y., Ma, L.: M4v: Multi-modal mamba for text-to-video generation. arXiv preprint arXiv:2506.10915 (2025)
18. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self forcing: Bridging the train-test gap in autoregressive video diffusion. arXiv preprint arXiv:2506.08009 (2025)
19. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al.: Vbench: Comprehensive benchmark suite for video generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21807–21818 (2024)
20. HunyuanWorld, T.: Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. arXiv preprint (2025)
21. HunyuanWorld, T.: Hy-world 1.5: A systematic framework for interactive world modeling with real-time latency and geometric consistency. arXiv preprint (2025)
22. Jacobs, S.A., Tanaka, M., Zhang, C., Zhang, M., Song, S.L., Rajbhandari, S., He, Y.: Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. arXiv preprint arXiv:2309.14509 (2023)
23. Jiang, H., Li, Y., Zhang, C., Wu, Q., Luo, X., Ahn, S., Han, Z., Abdi, A.H., Li, D., Lin, C.Y., Yang, Y., Qiu, L.: Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) Advances in Neural Information Processing Systems. vol. 37, pp. 52481–52515. Curran Associates, Inc. (2024). https://doi.org/10.52202/079017-1663
24. Kahatapitiya, K., Liu, H., He, S., Liu, D., Jia, M., Zhang, C., Ryoo, M.S., Xie, T.: Adaptive caching for faster video generation with diffusion transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15240–15252 (2025)
25. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)

26. Li, H., Shao, S., Zhong, W., Zhou, Z., Bai, L., Xiong, H., Xie, Z.: Pisa: Piecewise sparse attention is wiser for efficient diffusion transformers. arXiv preprint arXiv:2602.01077 (2026)
27. Li, M., Cai, T., Cao, J., Zhang, Q., Cai, H., Bai, J., Jia, Y., Li, K., Han, S.: Distrifusion: Distributed parallel inference for high-resolution diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7183–7193 (2024)
28. Li, M., Lin, Y., Zhang, Z., Cai, T., Li, X., Guo, J., Xie, E., Meng, C., Zhu, J.Y., Han, S.: Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. arXiv preprint arXiv:2411.05007 (2024)
29. Li, X., Gu, Y., Lin, X., Wang, W., Zhuang, B.: Psa: Pyramid sparse attention for efficient video understanding and generation. arXiv preprint arXiv:2512.04025 (2025)
30. Li, X., Li, M., Cai, T., Xi, H., Yang, S., Lin, Y., Zhang, L., Yang, S., Hu, J., Peng, K., Agrawala, M., Stoica, I., Keutzer, K., Han, S.: Radial attention: $o(n \log n)$ sparse attention with energy decay for long video generation (2025)
31. Li, Z., Li, H., Wu, J., Liu, K., Qin, H., Kong, L., Chen, G., Zhang, Y., Yang, X.: Dvd-quant: Data-free video diffusion transformers quantization. arXiv preprint arXiv:2505.18663 (2025)
32. Liang, C., Chen, H., Hou, L., Fan, Q., Wu, G., Tao, X., Wang, L.: Vmonarch: Efficient video diffusion transformers with structured attention. arXiv preprint arXiv:2601.22275 (2026)
33. Liu, F., Zhang, S., Wang, X., Wei, Y., Qiu, H., Zhao, Y., Zhang, Y., Ye, Q., Wan, F.: Timestep embedding tells: It's time to cache for video diffusion model. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 7353–7363 (2025)
34. Liu, H., Zaharia, M., Abbeel, P.: Ring attention with blockwise transformers for near-infinite context. arXiv preprint arXiv:2310.01889 (2023)
35. Liu, K., Hu, W., Xu, J., Shan, Y., Lu, S.: Rolling forcing: Autoregressive long video diffusion in real time. arXiv preprint arXiv:2509.25161 (2025)
36. Liu, Y., Hu, Y., Zhang, Z., Jiang, K., Yuan, K.: Mixture of distributions matters: Dynamic sparse attention for efficient video diffusion transformers. arXiv preprint arXiv:2601.11641 (2026)
37. Lu, C., Song, Y.: Simplifying, stabilizing and scaling continuous-time consistency models (2025)
38. Lv, C., Shi, Y., Huang, Y., Gong, R., Ren, S., Wang, W.: Light forcing: Accelerating autoregressive video diffusion via sparse attention. arXiv preprint arXiv:2602.04789 (2026)
39. Lv, Z., Si, C., Song, J., Yang, Z., Qiao, Y., Liu, Z., Wong, K.Y.K.: Fastercache: Training-free video diffusion model acceleration with high quality. arXiv preprint arXiv:2410.19355 (2024)
40. Ma, X., Fang, G., Bi Mi, M., Wang, X.: Learning-to-cache: Accelerating diffusion transformer via layer caching. Advances in Neural Information Processing Systems **37**, 133282–133304 (2024)
41. Ma, X., Fang, G., Wang, X.: Deepcache: Accelerating diffusion models for free. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15762–15772 (2024)
42. Ma, Z., Wei, L., Wang, F., Zhang, S., Tian, Q.: Magcache: Fast video generation with magnitude-aware cache. arXiv preprint arXiv:2506.09045 (2025)
43. Sun, W., Tu, R.C., Ding, Y., Jin, Z., Liao, J., Liu, S., Tao, D.: Vorta: Efficient video diffusion via routing sparse attention. arXiv preprint arXiv:2505.18809 (2025)

44. Sun, W., Zhang, H., Wang, H., Wu, J., Wang, Z., Wang, Z., Wang, Y., Zhang, J., Wang, T., Guo, C.: Worldplay: Towards long-term geometric consistency for real-time interactive world model. arXiv preprint (2025)
45. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
46. Wang, H., Ma, C.Y., Liu, Y.C., Hou, J., Xu, T., Wang, J., Juefei-Xu, F., Luo, Y., Zhang, P., Hou, T., et al.: Lingen: Towards high-resolution minute-length text-to-video generation with linear computational complexity. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 2578–2588 (2025)
47. Wu, J., Hou, L., Yang, H., Tao, X., Tian, Y., Wan, P., Zhang, D., Tong, Y.: Vmoba: Mixture-of-block attention for video diffusion models. arXiv preprint arXiv:2506.23858 (2025)
48. Wu, J., Wang, H., Shang, Y., Shah, M., Yan, Y.: Ptq4dit: Post-training quantization for diffusion transformers. Advances in neural information processing systems **37**, 62732–62755 (2024)
49. Wu, X., Wang, H., Zhou, Y., Lu, Q.: Usv: Unified sparsification for accelerating video diffusion models. arXiv preprint arXiv:2512.05754 (2025)
50. Xi, H., Yang, S., Zhao, Y., Li, M., Cai, H., Li, X., Lin, Y., Zhang, Z., Zhang, J., Li, X., et al.: Quant videogen: Auto-regressive long video generation via 2-bit kv-cache quantization. arXiv preprint arXiv:2602.02958 (2026)
51. Xi, H., Yang, S., Zhao, Y., Xu, C., Li, M., Li, X., Lin, Y., Cai, H., Zhang, J., Li, D., Chen, J., Stoica, I., Keutzer, K., Han, S.: Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity (2025)
52. Xia, Y., Ling, S., Fu, F., Wang, Y., Li, H., Xiao, X., Cui, B.: Training-free and adaptive sparse attention for efficient long video generation (2025)
53. Yang, S., Huang, W., Chu, R., Xiao, Y., Zhao, Y., Wang, X., Li, M., Xie, E., Chen, Y., Lu, Y., et al.: Longlive: Real-time interactive long video generation. arXiv preprint arXiv:2509.22622 (2025)
54. Yang, S., Xi, H., Zhao, Y., Li, M., Zhang, J., Cai, H., Lin, Y., Li, X., Xu, C., Peng, K., et al.: Sparse videogen2: Accelerate video generation with sparse attention via semantic-aware permutation. arXiv preprint arXiv:2505.18875 (2025)
55. Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., Freeman, B.: Improved distribution matching distillation for fast image synthesis. Advances in neural information processing systems **37**, 47455–47487 (2024)
56. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6613–6623 (2024)
57. Yin, T., Zhang, Q., Zhang, R., Freeman, W.T., Durand, F., Shechtman, E., Huang, X.: From slow bidirectional to fast autoregressive video diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22963–22974 (2025)
58. Zhang, H., Su, R., Yuan, Z., Chen, P., Shen, M., Fan, Y., Yan, S., Dai, G., Wang, Y.: Ditfastattnv2: Head-wise attention compression for multi-modality diffusion transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 16399–16409 (October 2025)
59. Zhang, J., Chen, M., Wang, H., Jiang, K., Stoica, I., Gonzalez, J.E., Chen, J., Zhu, J.: Sagebwd: A trainable low-bit attention. arXiv preprint arXiv:2603.02170 (2026)

60. Zhang, J., Huang, H., Zhang, P., Wei, J., Zhu, J., Chen, J.: Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization. In: International Conference on Machine Learning (ICML) (2025)
61. Zhang, J., Jiang, K., Xiang, C., Feng, W., Hu, Y., Xi, H., Chen, J., Zhu, J.: Spargeattention2: Trainable sparse attention via hybrid top-k+ top-p masking and distillation fine-tuning. arXiv preprint arXiv:2602.13515 (2026)
62. Zhang, J., Su, R., Liu, C., Wei, J., Wang, Z., Wang, H., Zhang, P., Jiang, H., Huang, H., Xiang, C., et al.: Efficient attention methods: Hardware-efficient, sparse, compact, and linear attention. `https://attention-survey.github.io/files/Attention_Survey.pdf` (2025), accessed: 27 June 2026
63. Zhang, J., Wang, H., Jiang, K., Yang, S., Zheng, K., Xi, H., Wang, Z., Zhu, H., Zhao, M., Stoica, I., et al.: Sla: Beyond sparsity in diffusion transformers via fine-tunable sparse-linear attention. arXiv preprint arXiv:2509.24006 (2025)
64. Zhang, J., Wang, H., Jiang, K., Zheng, K., Jiang, Y., Stoica, I., Chen, J., Zhu, J., Gonzalez, J.E.: Sla2: Sparse-linear attention with learnable routing and qat. arXiv preprint arXiv:2602.12675 (2026)
65. Zhang, J., Wei, J., Zhang, P., Xu, X., Huang, H., Wang, H., Jiang, K., Zhu, J., Chen, J.: Sageattention3: Microscaling fp4 attention for inference and an exploration of 8-bit training. arXiv preprint arXiv:2505.11594 (2025)
66. Zhang, J., Wei, J., Zhang, P., Zhu, J., Chen, J.: Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. In: International Conference on Learning Representations (ICLR) (2025)
67. Zhang, J., Xiang, C., Huang, H., Wei, J., Xi, H., Zhu, J., Chen, J.: Spargeattention: Accurate and training-free sparse attention accelerating any model inference. arXiv preprint arXiv:2502.18137 (2025)
68. Zhang, J., Xu, X., Wei, J., Huang, H., Zhang, P., Xiang, C., Zhu, J., Chen, J.: Sageattention2++: A more efficient implementation of sageattention2. arXiv preprint arXiv:2505.21136 (2025)
69. Zhang, P., Chen, Y., Huang, H., Lin, W., Liu, Z., Stoica, I., Xing, E., Zhang, H.: Vsa: Faster video diffusion with trainable sparse attention. arXiv preprint arXiv:2505.13389 (2025)
70. Zhao, T., Hong, K., Yang, X., Xiao, X., Li, H., Ling, F., Xie, R., Chen, S., Zhu, H., Zhang, Y., et al.: Paroattention: Pattern-aware reordering for efficient sparse and quantized attention in visual generation models. arXiv preprint arXiv:2506.16054 (2025)
71. Zheng, K., Wang, Y., Ma, Q., Chen, H., Zhang, J., Balaji, Y., Chen, J., Liu, M.Y., Zhu, J., Zhang, Q.: Large scale diffusion distillation via score-regularized continuous-time consistency. arXiv preprint arXiv:2510.08431 (2025)
72. Zhu, H., Zhao, M., He, G., Su, H., Li, C., Zhu, J.: Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. arXiv preprint arXiv:2602.02214 (2026)