INSTAGEN: GROUP-SHARED IMPORTANCE SAMPLING FOR FAST VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion Transformers (DiTs) are the dominant architecture for video generation, but their quadratic attention complexity results in substantial computational cost and latency. Permuted Block Sparse Attention offers a promising approach to reducing this cost by selectively computing blocks of a semantically permuted attention matrix. However, existing approaches typically select only the top few key blocks for each query block, following a top-p paradigm. We find that top-p selection can be highly sample inefficient in high-entropy attention regimes, which are common in video generation. In these regimes, we show that sampling can provide substantially better approximations under the same computational budget. More generally, we cast block selection as importance sampling, with top-k selection as the special case that samples no residual. Building on this formulation, we present InstaGen, a hardware-friendly importance sampling approach that produces structured sparsity amenable to efficient GPU execution while providing more accurate attention approximations than the standard top-p paradigm. On Wan2.1-T2V-1.3B and Wan2.1-T2V-14B, InstaGen improves PSNR over 3.7 dB and 1.2 dB, respectively, at matched attention density, and by 1.4 dB and 0.9 dB at matched wall-clock time. InstaGen also reaches SVG2's best quality 1.13x and 1.20x faster, respectively, while continuing to improve at attention densities where SVG2 has plateaued. Our anonymized code is available at: https://anonymous.4open.science/r/svg-research/.

1 INTRODUCTION

Diffusion Transformers (DiTs) have been instrumental in achieving state-of-the-art quality in image and video generation (Wan Team, 2025; Yang et al., 2025b; Kong et al., 2024). However, image and video generation with DiTs remains computationally expensive, primarily due to the quadratic complexity of the attention prefill workload, which is repeated across layers and diffusion steps. As a result, generating even short videos can be extremely slow. For instance, generating a single 5 second 720p video (81 frames, 50 denoising steps) with Wan2.1-T2V-14B takes about 22.6 minutes on one H100 GPU. Improving generation latency is therefore essential for enabling the efficient generation of high-quality, long-form videos.

Training-free Sparse Attention is a natural paradigm for reducing the computational workload of attention in existing video generation models and improving generation latency. However, effective attention sparsity must strike a unique balance between three often non-aligned goals: 1. Quality per query: Each query token should, in effect, attend to the tokens that are most important for that specific query. 2. Efficiency: The sparsity pattern should be amenable to block-based execution so that it can naturally integrate with FlashAttention (Shah et al., 2024) and benefit from hardware acceleration. 3. Alignment with the underlying model: While each query may, in principle, attend to its most important tokens, the approximation error introduced by sparsity can accumulate across layers and diffusion steps, potentially derailing the model and causing substantial downstream quality degradation. Prior methods such as SVG, SVG2, SVG-EAR (Xi et al., 2025; Yang et al., 2025a; Zhou et al., 2026) have primarily focused on addressing the first two challenges through algorithmic approaches. However, their treatment of the third challenge often results in an unfavorable trade-off between sparsity and downstream generation quality.

To identify the important tokens for each query, the goal is to estimate the corresponding attention scores. However, since computing per-query estimates incurs prohibitive overhead, this estimation

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

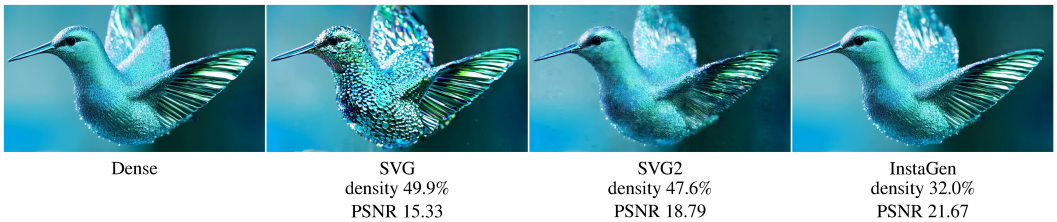


Figure 1: Frames from a single Wan2.1-T2V-1.3B generation (720p, 81 frames, 50 steps). InstaGen achieves higher pixel fidelity reaching a PSNR of 21.67 dB at a lower attention density of 32%.

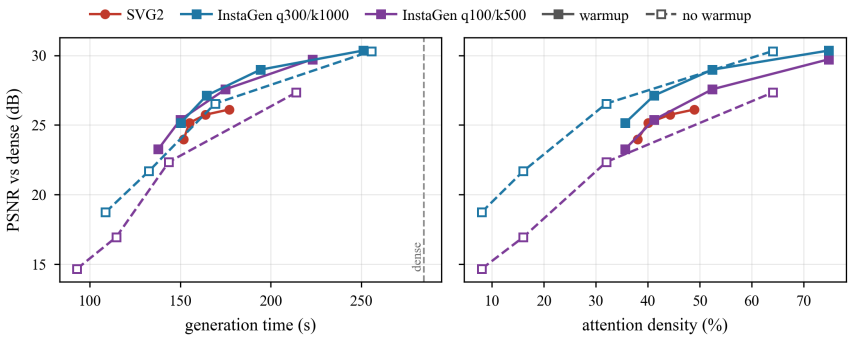


Figure 2: InstaGen gives higher quality than SVG2 at both matched density and matched latency on Wan2.1-T2V-1.3B. *Left:* PSNR against generation time per video. *Right:* PSNR against attention density, the share of attention computed over the whole generation. SVG2 uses the configuration from its paper (C_q=100, C_k=500, full attention for the first 30% of steps, key floor 0.1) at top-p 0.5, 0.75, 0.9 and 0.95. InstaGen runs exact + sampled budgets from 4+4% to 32+32% at two clustering configurations, with the same warmup and without it.

is performed at the block level. Specifically, a single vector represents a group of query vectors, and correspondingly, a single vector represents a group of key vectors. The resulting estimate for each group then serves as an estimate for all query-key pairs within the corresponding groups. Historically, these groups were formed based on token positions. This is problematic, especially when semantically diverse queries are grouped into a single cluster. More recently, SVG2 (Yang et al., 2025a) proposed clustering keys and queries into semantically aware groups before performing the attention estimation. This effectively addresses challenges (1) and (2). However, to ensure that attention-level approximation errors remain small, SVG2 employs a top-p paradigm, where the number of selected groups is dynamically chosen to cover a sufficient fraction of the attention mass. However, as shown in vAttention (Desai et al., 2026) for text models, top-p is not a sample-efficient paradigm for minimizing approximation error under a fixed token budget. With InstaGen, we instead shift from a top-p selection paradigm to a sampling-based paradigm, of which top-p selection becomes a special case.

In InstaGen, we formalize sparse attention as an importance sampling problem. We use attention score estimates produced by a scorer as importance sampling weights. Given a query q, let the resulting importance scores be $\mathbf{s} \in \mathbb{R}^d$. We perform proportional-to-size sampling without replacement (PPSWOR), first identifying the heavy-weighted tokens that should always be included. This naturally decomposes the sampled set into two components: a deterministic set of top-k tokens and a set of residual tokens selected through importance sampling. To enable efficient execution, we represent each query using its cluster centroid and perform sampling once for the centroid. All queries within the cluster can then share the resulting sampled token set, yielding a group-shared sampling scheme. This amortizes the sampling overhead across queries while producing structured sparsity that is compatible with FlashAttention and can be efficiently accelerated on GPUs.

We evaluate InstaGen against SVG, SVG2, and SVG-EAR on two video generation models, Wan2.1-T2V-1.3B and Wan2.1-T2V-14B. Across matched attention densities, InstaGen improves PSNR by

more than 3.5 dB, while achieving approximately 1-1.5 dB higher PSNR at matched generation latency. Importantly, InstaGen is the first approach to completely eliminate the dense warmup steps and perform sparse attention throughout all diffusion steps. Prior sparse video generation methods typically retain dense warmup steps to mitigate quality degradation caused by early sparse computation. By eliminating this requirement, InstaGen substantially reduces the computational cost of the entire generation process, rather than only accelerating the later diffusion steps. This property makes InstaGen particularly well suited to long-duration, high-quality video generation, where the cost of dense warmup steps becomes increasingly significant and the benefits of fully sparse generation compound with video length. On Wan2.1-T2V-1.3B, InstaGen without any warmup reaches 26.54 dB PSNR at 32% density, above SVG2's official warmup setting (26.11 dB at 48.9%), which keeps 30% of its steps dense.

2 Related Work

Structured sparse attention for video diffusion. Sparse VideoGen (SVG) profiles sampled query rows to assign each head a spatial or temporal sparsity layout (Xi et al., 2025), and Radial Attention uses a static $\mathcal{O}(S \log S)$ mask whose density decays with temporal distance (Li et al., 2025). Both fix the layout structurally rather than adapting it to content.

Content-adaptive sparse routing. Other methods estimate block importance from content: XAttention scores blocks by antidiagonal sums (Xu et al., 2025), AdaSpa searches hierarchical block patterns online (Xia et al., 2025), and SpargeAttention filters blocks with compressed query/key representatives and an in-kernel softmax-aware test (Zhang et al., 2025). SVG2 clusters queries and keys, permutes clusters into contiguous storage, and selects cluster pairs by a centroid score (Yang et al., 2025a); AdaCluster adapts the clustering to layerwise token geometry (Tan et al., 2026), and SVOO combines an offline per-layer sparsity schedule with online query–key co-clustering (Luo et al., 2026). These methods route each query cluster by a single centroid-derived score. InstaGen reuses this component to score token importance.

Recovering skipped attention. SVG-EAR approximates skipped blocks with key/value centroids and recomputes those with high predicted error (Zhou et al., 2026); SparsePR learns an affine residual correction from the computed attention (Taghavi et al., 2026); and Sol-Attn estimates the skipped contribution from the computed subset (Li et al., 2026). vAttention, developed for LLM decoding, instead samples the residual, giving an unbiased estimate with a probabilistic error guarantee (Desai et al., 2026). We extend this idea by casting sparse attention as importance sampling, under which top-k selection and residual sampling arise naturally.

Changing the model. Veda distills full-attention tile distributions into a learned selector (Han et al., 2026); SANA-Video trains a block-linear video diffusion transformer (Chen et al., 2026b), and SANA-Video 2.0 a hybrid with periodic softmax anchors and cross-depth attention residuals (Chen et al., 2026a). We focus on training-free sparse attention.

3 Background

3.1 Scaled Dot Product Attention

Consider one self-attention head with queries, keys, and values $Q, K, V \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the head dimension. For query i, the scaled attention logits are

$$\ell_{ij} = \frac{q_i^\top k_j}{\sqrt{d}}, \qquad j \in \{1, \dots, n\}. \tag{1}$$

Softmax attention then produces the output

$$o_i = \sum_{j=1}^{n} \alpha_{ij}\, v_j, \qquad \alpha_{ij} = \frac{e^{\ell_{ij}}}{\sum_{j'=1}^{n} e^{\ell_{ij'}}}, \tag{2}$$

or equivalently the ratio of two sums

$$o_i = \frac{N_i}{D_i}, \qquad N_i = \sum_{j=1}^{n} e^{\ell_{ij}} v_j, \qquad D_i = \sum_{j=1}^{n} e^{\ell_{ij}}. \tag{3}$$

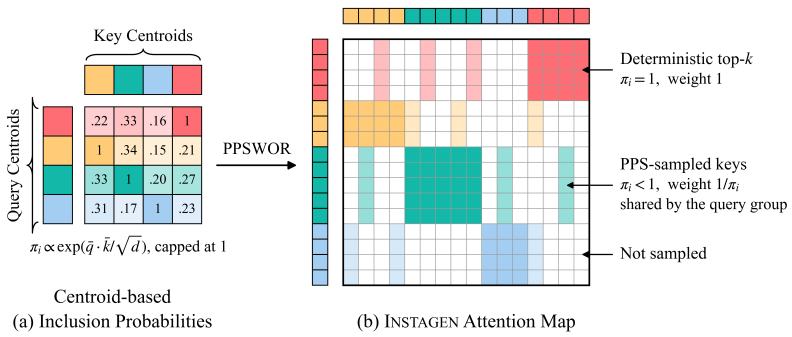


Figure 3: Visualization of an attention map computed via the InstaGen approach. PPSWOR leads to some key groups being included with probability 1, whereas other keys are sampled using the PPS algorithm and shared among the queries of a group. (Figure adapted from SVG2 (Yang et al., 2025a))

Evaluating the dense operator costs $O(n^2 d)$ per head. Sparse and sampled approximations target these sums: rather than including every key/value index j, they retain a small subset and estimate the contribution of the remainder. The sampling estimators below apply directly to approximating N_i and D_i.

3.2 Sampling with Probabilities Proportional to Size

Both N_i and D_i are sums over the n keys, so we consider the generic problem of estimating a sum

$$Y = \sum_{j=1}^{n} y_j \tag{4}$$

from a small sample of its terms. Setting $y_j = e^{\ell_{ij}} v_j$ recovers N_i, and setting $y_j = e^{\ell_{ij}}$ recovers D_i. Each term has a score $s_j > 0$, a cheap estimate of its magnitude $|y_j|$. The estimators below sample terms in proportion to these scores under a budget of R samples. Both are unbiased for any positive scores; the scores affect only the variance.

3.2.1 PPSWR: Probabilities Proportional to Size With Replacement

Sampling with replacement draws R independent indices $j_1, \ldots, j_R$, each equal to j with probability

$$p_j = \frac{s_j}{\sum_{j'=1}^{n} s_{j'}}. \tag{5}$$

The Hansen–Hurwitz estimator (Hansen & Hurwitz, 1943) is

$$\hat{Y} = \frac{1}{R} \sum_{r=1}^{R} \frac{y_{j_r}}{p_{j_r}}, \qquad \mathbb{E}(\hat{Y}) = Y, \qquad \mathbb{V}(\hat{Y}) = \frac{1}{R} \sum_{j=1}^{n} p_j \left(\frac{y_j}{p_j} - Y \right)^2. \tag{6}$$

The variance decays as $1/R$ and is zero when $p_j \propto |y_j|$, which is why the scores should track $|y_j|$.

3.2.2 PPSWOR: Probabilities Proportional to Size Without Replacement

Under PPSWR, a term with a large score can be drawn many times, and every repeated draw spends budget without covering a new term. Sampling without replacement avoids this by drawing a set $\mathcal{S}$ of R distinct terms. The design is then described by the inclusion probability of each term:

$$\pi_j = \Pr[\, j \in \mathcal{S} \,], \qquad 0 < \pi_j \leq 1, \qquad \sum_{j=1}^{n} \pi_j = R. \tag{7}$$

The Horvitz–Thompson estimator (Horvitz & Thompson, 1952) weights each sampled term by the inverse of its inclusion probability:

$$\hat{Y} = \sum_{j \in \mathcal{S}} \frac{y_j}{\pi_j}. \tag{8}$$

Algorithm 1 InstaGen attention for one head. $\ell_{ij} = q_i^\top k_j/\sqrt{d}$ are the logits of Section 3.1, $c(j)$ is the cluster of key j, and $|c|$ is its size. Budgets are fractions of the sequence: $e{+}s$ means $n_E = eN$ and $n_S = sN$. Warmup calls, when used, compute dense attention instead.

Input: $Q, K, V \in \mathbb{R}^{N\times d}$; budgets n_E, n_S; sampler UNIFORM or IMPORTANCE
Output: $\hat{O} \in \mathbb{R}^{N\times d}$

Cluster queries into C_q groups and keys into C_k clusters ▷ k-means, warm-started
$s_{gc} \leftarrow \exp\big(\bar{q}_g^\top \bar{k}_c/\sqrt{d}\big)$ for every pair (g, c) ▷ centroid scores
for each query group g **do**
  $\mathcal{E}_g \leftarrow$ whole clusters by decreasing $|c|\, s_{gc}$ until $|\mathcal{E}_g| \approx n_E$ ▷ exact keys
  $\mathcal{R}_g \leftarrow \{1, \dots, N\} \setminus \mathcal{E}_g$ ▷ residual keys
  if sampler is UNIFORM **then**
    $\pi_j \leftarrow n_S/|\mathcal{R}_g|$ for all $j \in \mathcal{R}_g$
    $\mathcal{S}_g \leftarrow n_S$ keys of $\mathcal{R}_g$, uniformly without replacement
  else
    $\pi_j \leftarrow \min\big(1, \lambda_g\, s_{g,c(j)}\big)$, with λ_g such that $\sum_{j\in\mathcal{R}_g} \pi_j = n_S$ ▷ §3.2.2
    $\mathcal{S}_g \leftarrow$ systematic PPS sample of $\mathcal{R}_g$ with probabilities π_j
  end if
  $w_j \leftarrow 1$ for $j \in \mathcal{E}_g$, $w_j \leftarrow 1/\pi_j$ for $j \in \mathcal{S}_g$ ▷ Horvitz–Thompson
  $\hat{o}_i \leftarrow \sum_{j\in\mathcal{E}_g\cup\mathcal{S}_g} \mathrm{softmax}_j\big(\ell_{ij} + \log w_j\big)\, v_j$ for all $i \in g$ ▷ $= \hat{n}_i/\hat{z}_i$
end for
return $\hat{O}$

The natural target $\pi_j = R\, s_j / \sum_{j'} s_{j'}$ exceeds 1 for any term whose score is more than a $1/R$ fraction of the total, but a term cannot be included more than once. Such a term is instead always included: it receives $\pi_j = 1$, contributes y_j exactly, and uses one unit of the budget. The inclusion probabilities are therefore

$$\pi_j = \min\left(1, \lambda s_j\right), \qquad \text{where } \lambda \text{ satisfies} \quad \sum_{j=1}^{n} \min\left(1, \lambda s_j\right) = R. \tag{9}$$

The left-hand side increases with λ, so λ is unique. The terms with $\pi_j = 1$ form the heavy set $\mathcal{H} = \{\, j : \lambda s_j \geq 1 \,\}$, which is the top-$k$ set by score, with $k = |\mathcal{H}|$ determined by the scores rather than fixed in advance.

The sample therefore consists of all of $\mathcal{H}$ and $R - |\mathcal{H}|$ residual terms, each included with probability exactly π_j. Separating the two parts of $\mathcal{S}$, the Horvitz–Thompson estimator becomes

$$\hat{Y} = \underbrace{\sum_{j\in\mathcal{H}} y_j}_{\text{deterministic top-}k} + \underbrace{\sum_{j\in\mathcal{S}\setminus\mathcal{H}} \frac{y_j}{\pi_j}}_{\text{sampled residual}}. \tag{10}$$

This estimator is unbiased. We use this formulation in InstaGen.

4 INSTAGEN

InstaGen builds on the observation in vAttention (Desai et al., 2026) that sampling can outperform top-p selection in high-entropy regimes. However, rather than adopting the uniform sampling strategy used in vAttention, InstaGen formulates sparse attention as a sampling problem, which naturally leads to an importance sampling formulation.

InstaGen performs PPSWOR sampling using scores that serve as proxies for the exponentiated raw attention scores. Consequently, InstaGen can be composed with existing methods such as SVG, SVG2, and SVOO, among others, since these methods all construct a proxy score for each query key pair. For clarity in our description and experiments, we adopt the clustering and semantic-aware permutation components of SVG2 (Yang et al., 2025a). We first cluster the keys and queries using the Lloyd-Max algorithm and then compute inner products between the resulting query and key cluster centroids. Let $\mathcal{C}(x)$ denote the centroid of the cluster containing vector x. We estimate the inner

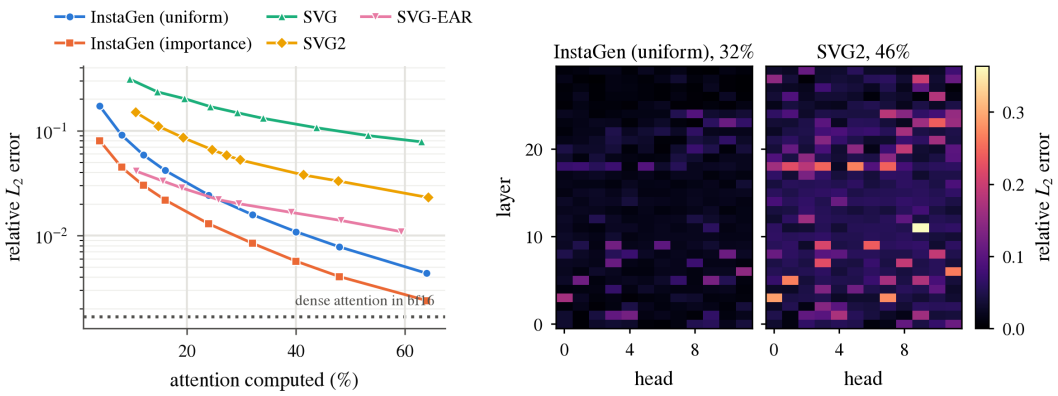


Figure 4: **InstaGen approximates attention more accurately than prior sparse methods.** *Left:* relative L_2 error of one self-attention call from Wan2.1-T2V-1.3B against exact attention. Importance sampling gives the lowest error at every density, several times below SVG2. *Right:* per-block, per-head error over one full generation averaged across all steps. InstaGen (uniform, 16+16, no warm-up) keeps the error low everywhere at 32% density, while SVG2 (p=0.9, warm-up counted as zero error) leaves large errors in individual heads and layers despite computing 46%.

product between a query and key by computing the inner product between the centroids of their respective clusters. Specifically, the score s_{ij} for query i and key j is defined as follows:

$$s_{ij} = e^{\langle \mathcal{C}(q_i), \mathcal{C}(k_j) \rangle}$$

These scores are used to perform PPSWOR as described in Section 3. Since all queries within a cluster share the same representative centroid vector, they also share the same scores. Thus, all of this computation is performed per query cluster. Similarly, since the keys are represented by their cluster centroids, the keys within a cluster have the same scores for a given query cluster. Since the top few keys have π set to 1 under the PPSWOR algorithm, this leads to the deterministic selection of some key clusters for each query cluster. The residual, i.e., the keys that are not selected deterministically, are sampled using PPS and shared across the queries. The resulting attention map is illustrated in Figure 3.

This sharing across queries is important for keeping the kernel GPU-friendly. Let S be the PPS sample over the residual. Under this sampling, the estimators for the numerator N and denominator D can be written as

$$\hat{D} = \sum_{i=1}^{n} \left(\mathbf{I}(\pi_i = 1) e^{\langle q, k_i \rangle} + \mathbf{I}(\pi_i < 1) \mathbf{I}(i \in S) \frac{1}{\pi_i} e^{\langle q, k_i \rangle} \right)$$

$$\hat{N} = \sum_{i=1}^{n} \left(\mathbf{I}(\pi_i = 1) e^{\langle q, k_i \rangle} \mathbf{v}_i + \mathbf{I}(\pi_i < 1) \mathbf{I}(i \in S) \frac{1}{\pi_i} e^{\langle q, k_i \rangle} \mathbf{v}_i \right)$$

where $\mathbb{I}(.)$ is the indicator function. Note that both the numerator and denominator estimators are unbiased:

$$\mathbb{E}(\hat{D}) = D \qquad \mathbb{E}(\hat{N}) = N.$$

Discussion on Sampling Alternatives and Their Possibilities There are various possibilities for sampling that we discuss below. While we use PPSWOR sampling in our kernel as an example, other possibilities also exist.

PPSWR. Sampling with replacement can be used in the algorithm above to estimate the residual. The downside of sampling with replacement is that we may obtain repeated samples. While this does not matter much statistically, it leads to undesirable properties at lower sparsities. For instance, it does not recover dense attention even when we use a budget equivalent to that of dense attention. On the other hand, sampling with replacement is easier to analyze analytically. The independence of samples allows us to use strong tail bounds, which can be used to devise a *verifiable* approximation of attention for sparse video generation. We leave these theoretical extensions as future work.

Table 1: **Main results on Wan2.1-T2V-1.3B and 14B.** E+S: exact and sampled share of keys per sparse call. Warm-up as prescribed by the official papers. Speedup is end-to-end on an H100 over dense (1.3B: 279.7 s; 14B: 1353.8 s).

			Dense-reference quality			VBench				Efficiency	
Method	E+S	Warm-up	PSNR↑	SSIM↑	LPIPS↓	ImgQual↑	AesQual↑	SubCons↑	BackCons↑	Density↓	Speedup↑
Wan2.1-T2V-1.3B											
Dense	–		–	–	–	0.6689	0.6416	0.9569	0.9612	100%	1.00×
SVG	–	✓	23.62	0.794	0.214	0.6837	0.6378	0.9489	0.9518	49.85%	1.47×
SVG2 (p=0.95)	–	✓	26.11	0.834	0.122	0.6185	0.6091	0.9420	0.9495	48.92%	1.65×
SVG-EAR	–	✓	28.29	**0.910**	0.064	0.6625	0.6357	0.9497	0.9543	34.87%	1.63×
InstaGen (uniform)	8+8	✓	27.12	0.856	0.093	0.6331	0.5798	0.9411	0.9467	41.20%	**1.76**×
InstaGen (uniform)	16+16		26.54	0.876	0.088	0.6718	**0.6490**	0.9521	0.9596	**32.00%**	1.68×
InstaGen (uniform)	16+16	✓	28.98	0.906	0.057	0.6434	0.5976	0.9528	0.9572	52.40%	1.49×
InstaGen (importance)	8+8	✓	27.69	0.869	0.071	0.6826	0.6272	0.9665	0.9591	41.20%	1.72×
InstaGen (importance)	16+16		27.72	0.891	0.061	**0.6909**	0.6312	**0.9702**	**0.9673**	**32.00%**	1.63×
InstaGen (importance)	16+16	✓	**29.39**	0.906	**0.053**	0.6864	0.6306	0.9701	0.9652	52.40%	1.46×
Wan2.1-T2V-14B											
Dense	–		–	–	–	0.6876	0.6676	0.9677	0.9715	100%	1.00×
SVG	–	✓	21.01	0.742	0.202	0.6892	**0.6742**	0.9653	0.9728	51.11%	1.41×
SVG2 (p=0.9)	–	✓	22.38	0.749	0.167	0.6927	0.6239	0.9661	0.9704	53.34%	1.53×
SVG-EAR	–	✓	23.55	0.827	0.126	0.6848	0.6700	0.9666	0.9677	42.19%	1.48×
InstaGen (uniform)	8+8	✓	22.91	0.771	0.136	0.6963	0.6345	**0.9763**	0.9693	42.67%	**1.67**×
InstaGen (uniform)	16+16		19.10	0.664	0.237	0.6792	0.6334	0.9736	**0.9748**	**32.00%**	1.63×
InstaGen (uniform)	16+16	✓	23.55	0.796	0.122	**0.6990**	0.6361	**0.9763**	0.9711	53.59%	1.45×
InstaGen (importance)	8+8	✓	**24.61**	**0.871**	**0.083**	0.6877	0.6701	0.9701	0.9701	42.67%	1.64×

PPSWOR. Sampling without replacement has several desirable properties for our algorithm. For instance, when operating at relatively low sparsity, PPSWOR is guaranteed to cover a larger number of tokens. In the limit, PPSWOR covers all tokens for each query and recovers the exact dense attention. However, the downside of using PPSWOR is the difficulty of analyzing the variance of the estimator and, consequently, obtaining analytical guarantees on the approximation error.

Poisson Sampling. Another sampling technique that can be used with the above algorithm for residual sampling is Poisson sampling. In Poisson sampling, we sample each token using a separate Bernoulli trial. The advantage is that we obtain a sample without replacement. However, the sample size is achieved only in expectation. Nevertheless, we can apply strong tail bounds to the estimators to obtain a *verifiable* sparse attention approximation even with Poisson sampling. Using different sampling schemes can be seen as extensions of this work and we leave this exploration for future.

5 Experiments

Models. We evaluate InstaGen on two open-source text-to-video diffusion transformers, Wan2.1-T2V-1.3B and Wan2.1-T2V-14B (Wan Team, 2025), generating 81-frame videos at 720p (720×1280). After 3D-VAE tokenization, each video has 21 latent frames of 3,600 tokens, so every self-attention call spans 75,600 tokens. We sparsify only self-attention; cross-attention to the text embeddings remains dense.

Metrics. We measure how closely each sparse method reproduces full attention with Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) and Learned Perceptual Image Patch Similarity (LPIPS). Each method is compared with the dense video from the same pipeline on the same hardware. We assess perceptual quality with four VBench (Huang et al., 2024) dimensions: Imaging Quality (ImgQual), Aesthetic Quality (AesQual), Subject Consistency (SubCons) and Background Consistency (BackCons). Efficiency is measured by *density*, the fraction of query-key pairs computed relative to full attention. We report it per sparse call and overall, where the overall figure counts dense warm-up calls as full attention. End-to-end efficiency is measured by wall-clock speedup over dense generation.

Datasets. We use 22 prompts from the VBench prompt suite (Huang et al., 2024), two chosen with most motion via an LLM as a judge from each of its eleven per-dimension prompt lists. Each prompt is generated once with a fixed seed, shared by all methods.

Baselines. We compare InstaGen against dense attention and three training-free sparse attention methods: the static method SVG (Xi et al., 2025), which assigns each head a spatial or temporal

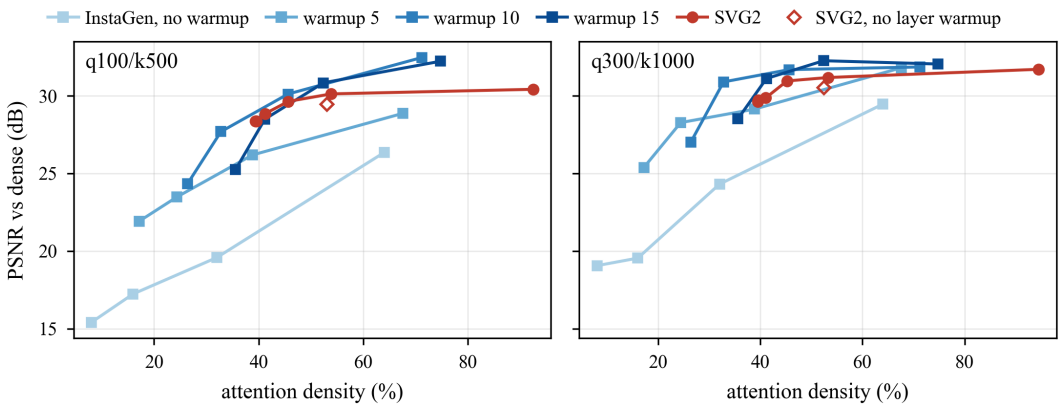


Figure 5: **Flooring ablations on Wan2.1-T2V-14B (720p, 81 frames, 50 steps).** InstaGen runs its first 0, 5, 10 or 15 of 50 steps at full attention (warmup), ranging from 4+4% to 32+32% budget. SVG2 runs at its paper configurations at top-p 0.1, 0.25, 0.5, 0.75, 0.9 and 0.99. A 5–10 step warmup matches or beats the 15 steps SVG2 uses. The hollow diamond is SVG2 without its one-layer warmup; the InstaGen runs have none.

pattern, and the dynamic methods SVG2 (Yang et al., 2025a), which keeps the k-means clusters that reach a top-p attention-mass threshold, and SVG-EAR (Zhou et al., 2026). Each baseline uses its official configuration and dense warm-up schedule. To isolate the effect of importance sampling, we also evaluate InstaGen with a *uniform* residual sampler. It is identical except that residual keys are drawn uniformly and weighted equally.

Implementation. We implement InstaGen with a fused attention kernel derived from FlashAttention-3 and benchmark on an NVIDIA H100 80GB SXM GPU. All methods use 50 UniPC steps with flow shift 5.0 and guidance scale 5.0. We set $C_q = 300$ and $C_k = 1000$, running 50 k-means iterations at a layer's first call and two warm-started iterations thereafter (Yang et al., 2025a). Each query group attends exactly to the top-scoring key blocks within $E\%$ of the keys and estimates the remainder from $S\%$ of the keys, sampled in proportion to their estimated attention mass (Section 3.2.2); we denote this E+S. Both terms are computed in one pass, with importance weights applied as a logit bias. Because InstaGen estimates the omitted attention rather than discarding it, it does not require the dense warm-up used by prior methods. We report it both with and without warm-up, with the first 30% of steps and 3% of layers dense, as in SVG2.

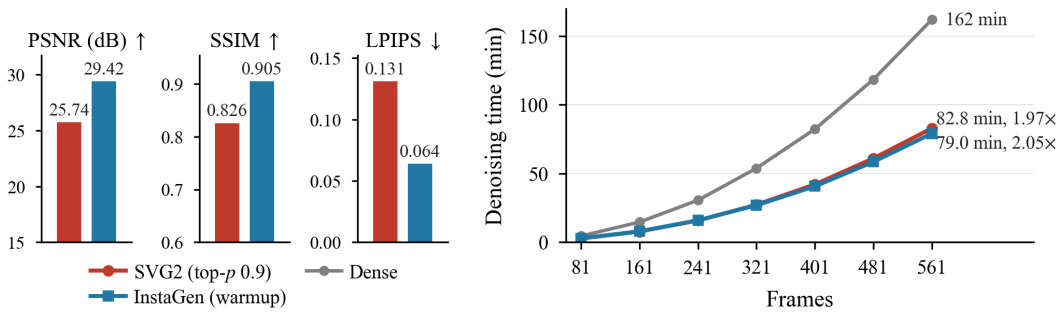
5.1 Quality and efficiency evaluation

Table 1 reports quality and efficiency across the two models. Both InstaGen variants share a route and a budget and differ only in the residual proposal: *uniform* draws the shared sample uniformly without replacement, while *importance* draws it proportionally to centroid affinity following PPSWOR.

On both models, InstaGen improves PSNR over SVG2 by 1.6 dB at 1.3B and 2.2 dB at 14B while using lower density and running faster. On the 14B model it is the best sparse method on all three fidelity metrics, ahead of SVG-EAR at nearly the same density. Without any warm-up, InstaGen still outperforms SVG and SVG2 on the 1.3B model, even though both run with warm-up. Importance sampling improves on uniform sampling at every configuration: drawing residual keys in proportion to their attention weight lowers the estimator's variance at the same budget.

Kernel efficiency. Sampled keys enter the kernel as a bias on a contiguous stream, so exact and sampled interactions are computed in a single online-softmax pass. Table 2 reports preparation separately from attention; InstaGen's preparation is costlier than SVG2's mainly because it uses more clusters (C_q=300, C_k=1000 vs. 100/500). Importance sampling only replaces the uniform sampler's constant weight with a per-key one. This slightly slows the attention kernel, but the draw itself is cheaper, so the two variants differ end to end by only a few percent.

Quality at matched density. We compare algorithms at similar density to get a fair comparison of attention approximation of every method. Figure 6a fixes the overall density at that of SVG2's



(a) Quality at matched density (~44.3%)

(b) Denoising time vs. video length

Figure 6: **At matched density, InstaGen gives much higher quality and runs as fast as SVG2 or faster.** (a) PSNR, SSIM and LPIPS at a matched overall density of 44.3%, averaged over 22 VBench prompts generated with Wan2.1-T2V-1.3B (720p, 81 frames, 50 steps). SVG2 uses its paper configuration; InstaGen takes the sparsified steps density (20.4%) and splits it equally between exact and sampled keys (10.2% + 10.2%), with the same 30% warmup. (b) Denoising time against the number of frames on one H100 SXM, at the same densities (VAE decoding excluded). As the number of frames grows, dense attention's cost grows fastest: at 561 frames it takes 162 min, against 82.8 min for SVG2 (1.97×) and 79.0 min for InstaGen (2.05×).

paper configuration (p=0.9, 44.3%) on Wan2.1-T2V-1.3B and gives InstaGen the same budget and warmup, split equally between exact and sampled keys (10.2+10.2). InstaGen raises PSNR from 25.74 to 29.42 dB and SSIM from 0.826 to 0.905, and lowers LPIPS from 0.131 to 0.064. Without any warmup, we can see in Figure 2, right that InstaGen lies above SVG2's entire frontier and on Wan2.1-T2V-14B InstaGen in (Figure 5) we see that it needs a short warmup, but 10 steps put it above SVG2 at every density.

Quality at matched latency. Figure 2 (left) plots quality against generation time on Wan2.1-T2V-1.3B. Without warmup, InstaGen (16+16) reaches 26.54 dB in 169.3 s, above SVG2's best setting (p=0.95, 26.11 dB in 177.0 s). With warmup, InstaGen (8+8) reaches 27.12 dB in 164.6 s, 1.0 dB higher than SVG2's best in 12.4 s less time. As videos get longer, latency increasingly favours InstaGen (Figure 6b). Table 2 (Appendix A) reports each method's latency on Wan2.1-T2V-14B: preparation, sampling and attention time per sparse call, end-to-end time, speedup over dense and peak memory.

Residual Approximation. Like InstaGen, SVG-EAR estimates the attention of skipped keys instead of dropping it, but deterministically. It replaces each skipped cluster with its centroid key and mean value. InstaGen instead samples actual skipped keys with inverse-probability weights, giving unbiased estimates of the attention sums whose variance falls as the sample grows.

6 Conclusion

This paper proposes InstaGen, a unified framework for token (or block) selection in sparse attention based on importance sampling. At its core, InstaGen formulates sparse attention as probability proportional to size sampling without replacement (PPSWOR): tokens with sufficiently high importance scores are deterministically retained, while the remaining attention is estimated by stochastic sampling proportional to their scores. This formulation unifies heavy-token selection and residual attention estimation within a single principled sampling procedure, rather than relying on a separate top-k selection rule and residual estimator.

Across sparse video generation experiments, InstaGen consistently achieves a better quality-efficiency tradeoff than existing methods. At matched density and latency, it produces higher-quality videos than top-k based methods and alternative residual estimation approaches. These results demonstrate that importance sampling provides an effective mechanism for allocating a limited attention budget between high-importance tokens and the residual attention mass, and offers a general framework that can be combined with different mechanisms for constructing proxy attention scores.

AI USE STATEMENT

We used AI tools to write and debug experiment and analysis code, to generate figures, to polish the writing, and for an initial exploration of the literature. We have reviewed all AI-assisted work. The authors are responsible for, and own, the ideas, explanations, experimental design and results presented in this work.

REPRODUCIBILITY STATEMENT

We have shared our code which can be used to reproduce the numbers on hardware specified in experiment section.

REFERENCES

Junsong Chen, Jincheng Yu, Yitong Li, Shuchen Xue, Haozhe Liu, Jingyu Xin, Yuyang Zhao, Tian Ye, Zhangjie Wu, Zian Wang, Daquan Zhou, Ping Luo, Song Han, and Enze Xie. SANA-Video 2.0: Hybrid linear attention with attention residuals for efficient video generation, 2026a. URL https://arxiv.org/abs/2607.21553.

Junsong Chen, Yuyang Zhao, Jincheng Yu, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Daquan Zhou, 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 *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2509.24695. Oral.

Aditya Desai, Kumar Agrawal, Shuo Yang, Alejandro Cuadron Lafuente, Luis Gaspar Schroeder, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. vAttention: Verified sparse attention via sampling. In *International Conference on Learning Representations*, pp. 52253–52284, 2026. arXiv:2510.05688.

Shihao Han, Hao Yang, Xiaofeng Mei, Xinting Hu, Yi Jiang, and Xiaojuan Qi. Veda: Scalable video diffusion via distilled sparse attention. In *International Conference on Machine Learning*, 2026.

Morris H. Hansen and William N. Hurwitz. On the theory of sampling from finite populations. *The Annals of Mathematical Statistics*, 14(4):333–362, 1943.

D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.

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*, pp. 21807–21818, 2024.

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.

Haopeng Li, Yitong Li, Junsong Chen, Tian Ye, Haozhe Liu, Jincheng Yu, Duomin Wang, Ruihua Zhang, Zeke Xie, Enze Xie, and Song Han. Sol-Attn: Accelerating video generation inference via on-the-fly attention sparsification, 2026. URL https://arxiv.org/abs/2607.24027. Technical report.

Xingyang Li, Muyang Li, Tianle Cai, Haocheng Xi, Shuo Yang, Yujun Lin, Lvmin Zhang, Songlin Yang, Jinbo Hu, Kelly Peng, Maneesh Agrawala, Ion Stoica, Kurt Keutzer, and Song Han. Radial attention: $\mathcal{O}(n \log n)$ sparse attention with energy decay for long video generation. *arXiv preprint arXiv:2506.19852*, 2025.

Jiayi Luo, Jiayu Chen, Jiankun Wang, Cong Wang, Hanxin Zhu, Qingyun Sun, Chen Gao, Zhibo Chen, and Jianxin Li. Attention sparsity is input-stable: Training-free sparse attention for video generation via offline sparsity profiling and online QK co-clustering, 2026. URL https://arxiv.org/abs/2603.18636. Accepted to ICML 2026.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2193.

Pardis Taghavi, Reza Langari, and Gaurav Pandey. Partition the support, reconstruct the residual: Training-free sparse attention for video generation and world models, 2026. URL https://arxiv.org/abs/2608.18484.

Haoyue Tan, Shengnan Wang, Yulin Qiao, Juncheng Zhang, Youhui Bai, Ping Gong, Zewen Jin, and Cheng Li. AdaCluster: Adaptive query-key clustering for sparse attention in video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43249–43259, 2026.

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

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68208–68224, 2025.

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.

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

Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, Jianfei Chen, Song Han, Kurt Keutzer, and Ion Stoica. Sparse VideoGen2: Accelerate video generation with sparse attention via semantic-aware permutation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 96965–96991, 2025a. doi: 10.52202/085713-3243.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025b.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. SpargeAttention: Accurate and training-free sparse attention accelerating any model inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76397–76413, 2025.

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. URL https://arxiv.org/abs/2603.08982. Accepted to ECCV 2026.

A SUPPLEMENTARY MATERIAL

Table 2: **Latency breakdown on Wan2.1-T2V-14B**. Prepare, sample and attention are milliseconds per sparse call, averaged over the generation. Prepare includes sampling, and for SVG-EAR its compensation step. Attention is the kernel alone. Speedup is end-to-end over dense. Peak memory is measured during denoising. The baselines run in their official pipeline.

Method	$E{+}S$	Warm-up	Density (%)	Prepare (ms)	Sample (ms)	Attention (ms)	Transformer (s)	E2E (s)	Speedup	Peak mem. (GiB)
Dense (cuDNN)	–		100	–	–	201.8	1350.8	1356.8	1.00×	56.0
SVG	–	✓	51.11	9.3	–	130.9	948.9	955.4	1.42×	46.6
SVG2	–	✓	54.52	24.9	–	79.7	875.9	881.9	1.54×	51.3
SVG-EAR	–	✓	42.49	56.6	–	72.1	908.9	915.0	1.48×	53.5
InstaGen (uniform)	8+8	✓	42.67	40.6	2.9	42.5	806.7	812.9	1.67×	53.2
InstaGen (uniform)	16+16		32.00	41.3	6.0	84.8	825.0	831.4	1.63×	53.2
InstaGen (uniform)	16+16	✓	53.59	42.8	6.0	84.8	931.5	937.8	1.45×	53.2
InstaGen (importance)	8+8	✓	42.67	39.7	2.2	48.0	816.9	823.1	1.65×	53.2
InstaGen (importance)	16+16		32.00	39.3	3.3	94.9	852.6	858.9	1.58×	53.2
InstaGen (importance)	16+16	✓	53.59	40.8	3.3	94.8	950.5	956.7	1.42×	53.2