ESTRA: ENABLING EXTREME ATTENTION SPARSITY FOR TRAINING-FREE VIDEO DIFFUSION ACCELERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video diffusion transformers generate high-quality videos, but attention over long spatiotemporal sequences makes inference computationally expensive. Sparse attention reduces this cost by computing only a subset of query–key interactions exactly, and recent training-free methods further approximate the remaining interactions. These methods, however, degrade quickly at high sparsity, which limits the achievable speedup. We present ESTRA (Extreme Sparsity Through Refined Attention), a training-free acceleration method that preserves generation quality even at extreme sparsity. ESTRA concentrates exact attention computation on a small set of important key–value tokens and approximates the remaining contributions with cluster representatives. To identify these tokens efficiently, it groups similar queries and uses their representatives to score individual keys. A variance-based correction further refines the attention weights of the cluster representatives, improving the accuracy of the approximation. Experiments on Wan2.1-14B and HunyuanVideo-13B show that ESTRA establishes a new quality–efficiency frontier for training-free sparse attention, **achieving** $3.26\times$ **and** $4.71\times$ **end-to-end speedups at 5% attention density** while surpassing Sol-Attn and SVG-EAR in fidelity to dense generation and on most VBench dimensions. **Even at 1% density**, ESTRA remains more faithful to dense generation than Sol-Attn at 12.5%, while reaching up to $5.39\times$ speedup.

1 INTRODUCTION

Diffusion and flow-based models support a range of video tasks, from generation conditioned on text or images to editing existing footage (Bar-Tal et al., 2024; Polyak et al., 2024). Many of these models rely on transformers to capture spatial and temporal relationships across long token sequences (Ma et al., 2025; Yang et al., 2025b; Polyak et al., 2024). Full attention scales quadratically with sequence length, making longer and higher-resolution videos particularly expensive to generate. Repeated across sampling steps, attention accounts for a substantial part of the cost of generation.

Several lines of work reduce inference cost without retraining the model. High-order solvers require fewer network evaluations (Lu et al., 2022), caching reuses intermediate computations across sampling steps (Liu et al., 2025; Lv et al., 2025), and quantization reduces the arithmetic cost within each evaluation (Zhang et al., 2025a). For attention itself, efficient exact kernels such as FlashAttention reduce memory traffic but still evaluate all query–key interactions (Dao et al., 2022). Sparse attention aims to approximate dense attention outputs at lower computational cost by evaluating only a subset of query–key interactions exactly.

Video attention often assigns most of its weight to a small fraction of tokens (Yang et al., 2025a). Sparse methods identify important interactions using spatial and temporal patterns (Xi et al., 2025; Zhang et al., 2025c) or semantic clusters (Yang et al., 2025a). Even individually weak interactions can collectively contribute to the output, motivating recent methods to estimate the unselected contributions as well. SVG-EAR uses cluster means for compensation and estimated compensation errors to guide block selection (Zhou et al., 2026). Sol-Attn reuses scores between queries and pooled keys for both block selection and compensation within a single attention kernel (Li et al., 2026a).

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.

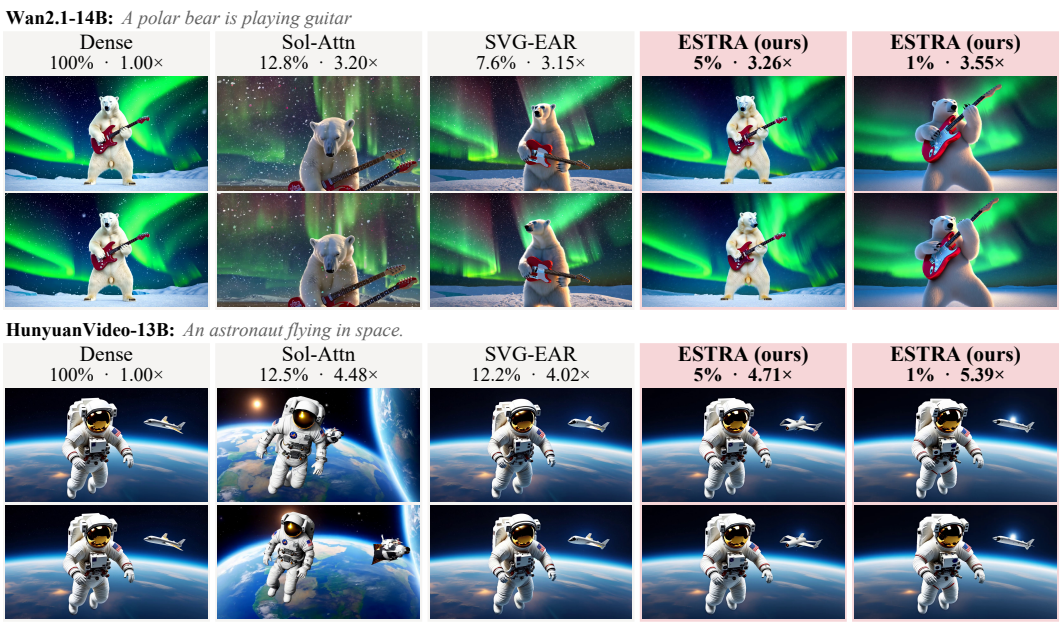

Figure 1: **High-quality video generation at up to 99% attention sparsity.** At 5% exact-interaction density, ESTRA more closely preserves the composition and appearance of the dense references than the baselines shown, while using higher sparsity and achieving greater end-to-end speedups. At 1%, the videos remain visually coherent, with particularly small changes in the HunyuanVideo example. Density denotes the fraction of video query–key interactions evaluated exactly; speedups are end to end on an NVIDIA H20. Additional examples are provided in Appendix D.

By combining selection with compensation, these methods can preserve generation quality at sparsities approaching 90% (Li et al., 2026a; Zhou et al., 2026). Preserving this quality at more extreme sparsity, however, becomes more challenging as the remaining budget for exact computation shrinks. For example, increasing sparsity from 90% to 99% reduces the fraction of query–key interactions computed exactly from 10% to just 1%. This represents *a further tenfold reduction in exact computation*—the same relative reduction as moving from dense attention to 90% sparsity.

We introduce ESTRA (Extreme Sparsity Through Refined Attention), a training-free acceleration method for video diffusion transformers. ESTRA further reduces approximation error through *finer token selection* and *more accurate cluster compensation*, enabling high-quality generation at up to 99% attention sparsity. Clustering similar queries lets them share selection scores, while clustering similar keys provides compact key–value representatives. ESTRA uses each query centroid to score every key, so the limited budget can be assigned to high-scoring tokens across clusters. Sharing these scores keeps token-level selection inexpensive.

For the remaining contributions, count-weighted cluster means provide an inexpensive approximation, but omit score variation within each cluster. This variation matters because softmax exponentiates scores before normalization. Motivated by a second-order expansion, we develop variance-calibrated mean (VCM) compensation to refine representative weights using the dispersion of the remaining keys. Each query jointly attends to selected tokens and corrected cluster representatives, reducing attention-output error at extreme sparsity.

Our contributions are summarized as follows:

- We propose ESTRA, a training-free sparse attention method for accelerating video diffusion transformers. It selects individual keys for exact computation using query-centroid scores shared within each query cluster, and approximates the remaining contributions with cluster representatives whose weights are refined by VCM. Together, these components closely approximate dense attention outputs at high sparsity.

- Experiments on Wan2.1-14B and HunyuanVideo-13B show that ESTRA establishes a new quality–speed frontier for training-free sparse attention. **At only 5% density**, it surpasses

Sol-Attn and SVG-EAR at higher densities in fidelity to dense generation and on most evaluated VBench dimensions, with larger end-to-end speedups of $3.26\times$ and $4.71\times$.

- To our knowledge, ESTRA is the first training-free sparse attention method to generate high-quality videos at **99% attention sparsity**, where existing methods either cannot operate or produce severe artifacts. At 1% density, it reaches end-to-end speedups of $3.55\times$ and $5.39\times$ on Wan2.1-14B and HunyuanVideo-13B, while remaining more faithful to dense generation than Sol-Attn at 12.5%.

2 Related Work

Training-free video diffusion acceleration. High-order solvers such as DPM-Solver reduce the number of network evaluations needed for diffusion sampling (Lu et al., 2022). Caching methods such as TeaCache and FasterCache reuse or predict intermediate outputs across denoising steps, reducing repeated model computation (Liu et al., 2025; Lv et al., 2025). D2Cache improves prediction accuracy using second-order residual deltas, with adaptive scaling guided by timestep embeddings to limit error accumulation (Liu et al., 2026a). At the operator level, SageAttention uses low-precision matrix multiplications (Zhang et al., 2025a), and the dense FlashAttention kernel reduces memory traffic while computing exact attention (Dao et al., 2022). ESTRA focuses on reducing the query–key interactions evaluated exactly within each attention call, retaining an approximation of the remaining contributions.

Sparse attention for video generation. Sparse methods differ in how they identify important interactions and organize them for efficient execution. Sparse VideoGen exploits the spatial or temporal preferences of attention heads (Xi et al., 2025), and Sliding Tile Attention maps local spatiotemporal windows to GPU-friendly tiles (Zhang et al., 2025c). Content-dependent methods estimate block importance from the current queries and keys: SpargeAttention uses similarity-aware pooling and online softmax filtering (Zhang et al., 2025b), whereas XAttention uses antidiagonal sampling (Xu et al., 2025). For near-lossless acceleration at moderate sparsity, LoSA selects blocks to retain 99% of the attention mass measured at an early dense step and reuses their indices thereafter (Liu et al., 2026b). SVG2 forms semantic groups by clustering and permuting queries and keys, then selects key clusters using centroid-based estimates (Yang et al., 2025a). Chipmunk combines column-level selection with cached attention updates across diffusion steps (Silveria et al., 2025).

Compensating for unselected interactions. Recent methods combine sparse exact computation with an estimate of the unselected contributions. PISA combines exact attention on selected blocks with Taylor approximations of the remaining contributions under a shared softmax normalization (Li et al., 2026b). SVG-EAR uses key–value cluster means for compensation and guides exact block selection using estimates of compensation error per unit cost (Zhou et al., 2026). Its query-centroid probes estimate per-key compensation errors, which are aggregated for block selection. Sol-Attn reuses scores between queries and pooled keys for block selection and compensation, integrating both with attention evaluation in a single kernel (Li et al., 2026a). For each query cluster, ESTRA selects individual keys for exact computation and approximates the remaining contributions with cluster representatives. VCM refines representative weights using the remaining keys' dispersion.

3 Preliminaries

Attention. For an attention head with queries $Q \in \mathbb{R}^{N_q \times d}$ and keys and values $K, V \in \mathbb{R}^{N_k \times d}$, the output is

$$O = \mathrm{softmax}(S)V, \qquad S = \frac{QK^\top}{\sqrt{d}}, \tag{1}$$

where softmax is applied row-wise and $s_{ij} = q_i^\top k_j/\sqrt{d}$ denotes an individual attention logit. Evaluating all query–key pairs costs $\mathcal{O}(N_q N_k d)$. FlashAttention avoids storing the full attention matrix through tiling and online softmax, but still evaluates every query–key pair (Dao et al., 2022).

Sparse attention. Sparse attention saves computation by retaining selected query–key pairs:

$$O_{\mathrm{sparse}} = \mathrm{softmax}(S + M)V, \qquad M \in \{0, -\infty\}^{N_q \times N_k}. \tag{2}$$

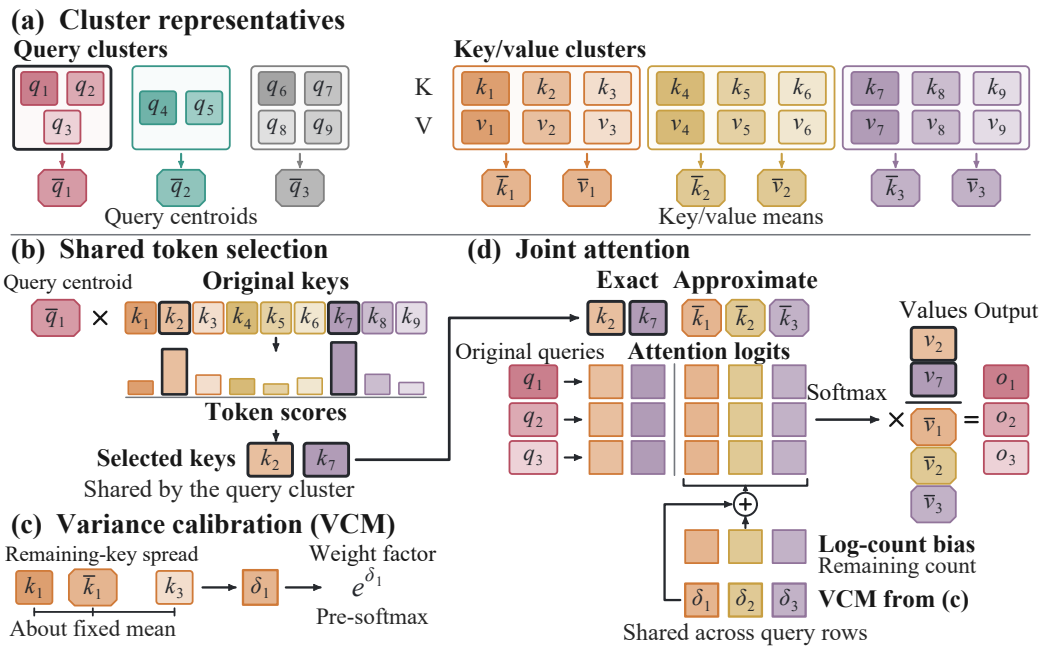

Figure 2: ESTRA overview. **(a)** Independent query and key clusters provide query centroids and K/V means. **(b)** Each centroid scores individual keys for selection shared by its query cluster. **(c)** VCM uses remaining-key spread and query norms to calibrate representative weights. **(d)** Original queries attend jointly to selected tokens and fixed cluster means, with log-count and VCM biases on representative logits. Colors identify clusters. Attention scaling is omitted.

Interactions with $M_{ij} = -\infty$ are skipped; the fraction with $M_{ij} = 0$ is the attention density. For efficient GPU execution, methods often select entire query–key blocks and skip the rest (Yang et al., 2025a; Xu et al., 2025). All queries in one block then attend to the same selected key blocks. Block importance can be estimated inexpensively, for example through dot products between block means.

Compensating for unselected interactions. Although sparse attention reduces computation, it also discards blocks whose combined contribution to the output may be substantial. Recent methods therefore add compensation to approximate these omitted contributions alongside exact computation on the selected tokens (Li et al., 2026b; Zhou et al., 2026; Li et al., 2026a). A common approach represents each unselected block by its mean key and mean value. For a block of queries, let S_{sel} contain the logits for selected keys and S_{rep} those for the representatives. The corresponding value matrices contain the original values, V_{sel}, and block-mean values, V_{rep}. The row vector n_{rep} records how many tokens each representative stands for. The two contributions are combined as

$$\widetilde{O} = \text{softmax}([\, S_{\text{sel}},\ S_{\text{rep}} + \log n_{\text{rep}} \,]) \begin{bmatrix} V_{\text{sel}} \\ V_{\text{rep}} \end{bmatrix}. \tag{3}$$

The log-count bias is shared across query rows. Adding $\log n$ to a representative's logit multiplies its unnormalized weight by n, the number of tokens it represents. A single softmax jointly normalizes selected-token and representative weights; both sets of logits use the original queries.

Sol-Attn applies mean-based compensation to fixed-size blocks (Li et al., 2026a). For each query block, it selects key blocks whose proxy scores exceed a threshold set from the mean and standard deviation of these scores. It reuses scores between queries and mean keys for selection and compensation, and fuses these operations with exact attention in a single kernel.

Clustering for selection and compensation. The quality of selection and compensation also depends on how blocks are formed. Tokens adjacent in the sequence can have very different attention patterns. SVG2 and SVG-EAR instead group queries and keys by feature similarity, clustering them independently and reordering tokens by cluster assignment (Yang et al., 2025a; Zhou et al., 2026). The resulting layout concentrates high-weight interactions into denser blocks of the attention map.

At a given density, selected blocks can then retain more of the dense attention weight, improving attention recall and helping reduce approximation error (Yang et al., 2025a).

Grouping similar keys also improves the mean approximation used for compensation (Zhou et al., 2026). SVG-EAR combines this clustered representation with error-aware selection. For each query cluster, it uses the centroid to estimate the compensation error for each key. It sums these estimates within each key cluster and ranks clusters by estimated error per unit of computation. The selected clusters are evaluated exactly; the rest use mean representatives in the joint softmax of Eq. (3).

4 METHOD

ESTRA builds on this clustered representation to improve both compensation and selection. We first examine how mean-based compensation estimates the attention weight of an unselected cluster. For a given query, the logit computed from a cluster's mean key equals the average of its members' logits. The cluster's total unnormalized attention weight, however, is the sum of the exponentials of their individual logits. Multiplying the mean key's weight by the cluster size, as in Eq. (3), does not generally recover this sum. We introduce variance-calibrated mean (VCM) compensation, which uses variation among the keys to refine this weight estimate.

We then refine selection to use the exact-computation budget more effectively. Under whole-cluster selection, retaining a high-scoring key also requires evaluating the rest of its cluster. We instead use each query centroid to score individual keys, allowing high-scoring members from different clusters to be selected together. Scoring once per query cluster keeps this finer selection inexpensive. Figure 3 shows each refinement's error reduction alongside Sol-Attn and SVG-EAR results.

Clustered representation. Following SVG2 and SVG-EAR, we apply k-means independently to Q and K (Yang et al., 2025a; Zhou et al., 2026). This yields C_q query clusters $\mathcal{G}_g$ and C_k key clusters $\mathcal{C}_c$. We reorder queries and keys by cluster assignment; values follow their corresponding keys. The query centroids $\bar{q}_g$ guide selection, and the K/V means $(\bar{k}_c, \bar{v}_c)$ computed over each full key cluster serve as compensation representatives (Figure 2(a)). Let T_g denote the selected key indices shared by all queries in $\mathcal{G}_g$. We first derive compensation for a given T_g, whether it contains whole clusters or individual tokens. We then describe how query centroids select individual keys.

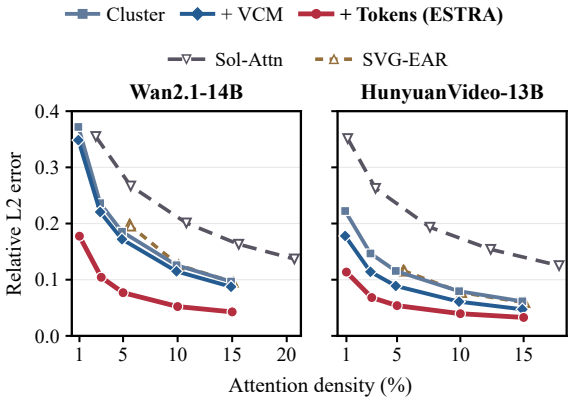

Figure 3: Adding VCM and then token-level selection lowers attention-output error on both backbones.

Variance-calibrated mean compensation. VCM improves compensation by adjusting the representative weight to reflect variation among the remaining keys.

For query cluster g, the unselected members of key cluster c are $\mathcal{R}_{g,c} = \mathcal{C}_c \setminus T_g$, with $r_{g,c} = |\mathcal{R}_{g,c}|$ remaining members. Different query clusters may leave different keys unselected, so recomputing their K/V means would require separate representatives for each query–key cluster pair. We instead share the full-cluster means $(\bar{k}_c, \bar{v}_c)$ across query clusters and adjust their weights to account for the remaining tokens.

For a query $i \in \mathcal{G}_g$ and a nonempty remainder $\mathcal{R}_{g,c}$, let $z_{i,c} = q_i^\top \bar{k}_c / \sqrt{d}$ be the representative logit. The logit deviation of a remaining key is $\epsilon_{ij} = s_{ij} - z_{i,c}$. Expectations and variances below are taken over the remaining keys, with equal weight for each key. Their exact total unnormalized weight is

$$M_{i,g,c} = \sum_{j \in \mathcal{R}_{g,c}} e^{s_{ij}} = r_{g,c}\, e^{z_{i,c}}\, \mathbb{E}[e^{\epsilon_{ij}}]\,. \tag{4}$$

Mean compensation uses only $r_{g,c}e^{z_{i,c}}$, omitting the factor $\mathbb{E}[e^{\epsilon_{ij}}]$. For small logit deviations, expanding the logarithm of this factor to second order gives two contributions:

$$\log \mathbb{E}[e^{\epsilon_{ij}}] \approx \underbrace{\mathbb{E}[\epsilon_{ij}]}_{\text{mean shift}} + \underbrace{\tfrac{1}{2}\operatorname{Var}(\epsilon_{ij})}_{\text{dispersion}}. \tag{5}$$

VCM uses a scalar proxy for the dispersion term and omits the first-order mean shift. We measure the spread of the remaining keys about the fixed full-cluster center (Figure 2(c)):

$$m_{2,g,c} = \frac{1}{r_{g,c}} \sum_{j \in \mathcal{R}_{g,c}} \|k_j - \bar{k}_c\|_2^2. \tag{6}$$

Selection can shift the remaining keys' mean away from $\bar{k}_c$, so $m_{2,g,c}$ measures spread about the fixed cluster center rather than the remaining mean. We use this statistic to estimate the second moment of the logit deviations, $\mathbb{E}[\epsilon_{ij}^2]$, as a proxy for their variance. Under an isotropic approximation for the key deviations, $\frac{1}{2}\mathbb{E}[\epsilon_{ij}^2] \approx \|q_i\|_2^2 m_{2,g,c}/(2d^2)$. We use one scale α per attention head by averaging $\|q_i\|_2^2/(2d^2)$ over its queries. Multiplying this scale by $m_{2,g,c}$ and applying a cap $\delta_{\max}$ gives the correction:

$$\alpha = \frac{1}{2d^2 N_q} \sum_{i=1}^{N_q} \|q_i\|_2^2, \qquad \delta_{g,c} = \min(\alpha m_{2,g,c}, \delta_{\max}), \tag{7}$$

The cap $\delta_{\max} \geq 0$ prevents the correction from increasing representative weights by more than a factor of $e^{\delta_{\max}}$.

Adding this correction to the mean-key logit and the remaining-token count bias gives

$$\ell_{i,c} = z_{i,c} + \log r_{g,c} + \delta_{g,c}, \qquad r_{g,c} > 0. \tag{8}$$

This replaces the omitted factor $\mathbb{E}[e^{\epsilon_{ij}}]$ in Eq. (4) with $e^{\delta_{g,c}}$. We obtain the remaining count and squared-deviation sum by subtracting selected-token statistics from full-cluster totals. This updates only two scalars per query–key cluster pair, without recomputing K/V means. Empty remainders are omitted.

With selection and K/V means unchanged, VCM lowers our cluster-level baseline's error throughout the density sweep in Figure 3. At 5% target density, the reductions are 7.2% on Wan and 23.1% on HunyuanVideo.

Token-level selection. VCM refines the estimated contribution of unselected keys. We now turn to selecting the keys that receive exact computation. Selecting a whole cluster includes its lower-scoring members as well as its important ones. Selecting individual keys instead lets the same budget cover high-scoring tokens from multiple clusters. We preserve the cluster representatives to compensate for the remaining tokens.

Query clustering makes this finer selection inexpensive. We score each key by its mean logit across the queries in cluster g:

$$a_{gj} = \frac{\bar{q}_g^\top k_j}{\sqrt{d}} = \frac{1}{|\mathcal{G}_g|} \sum_{i \in \mathcal{G}_g} s_{ij}. \tag{9}$$

Scoring once per query cluster requires $C_q N_k$ dot products, compared with $N_q N_k$ for per-query scoring. Figure 2(b) illustrates this shared selection: $\bar{q}_1$ scores the original keys, and k_2 and k_7 are retained from different clusters.

For a target density ρ, we aim to retain the highest-scoring ρ fraction of keys. We approximate this selection with a threshold estimated from sampled key scores, avoiding full-sequence sorting and score storage. The selected indices form T_g: their original keys and values receive exact attention, while the remaining tokens are handled by compensation.

Keeping VCM and replacing whole-cluster selection with token-level selection further reduces error across the sweep in Figure 3. At a 5% target density, the reductions are 55.3% on Wan and 39.4% on HunyuanVideo. The finer selection thus makes better use of the exact-attention budget while retaining compact compensation for the remaining tokens.

Joint attention. Each original query attends jointly to the selected tokens and calibrated cluster representatives (Figure 2(d)). The joint softmax in Eq. (3) now uses $\log r_{g,c} + \delta_{g,c}$ as the bias for each representative. A representative is omitted if all of its keys have been selected. For fixed cluster counts, token scoring and representative attention scale linearly with sequence length.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models and baselines. We evaluate ESTRA on Wan2.1-14B and HunyuanVideo-13B (Wan Team, 2025; Kong et al., 2024), generating 81- and 129-frame videos at 1280×720 resolution, respectively. We compare against dense attention and two state-of-the-art training-free sparse attention methods, Sol-Attn (Li et al., 2026a) and SVG-EAR (Zhou et al., 2026). We choose these baselines for their strong quality–speed trade-offs: SVG-EAR achieves low approximation error, while Sol-Attn provides a highly efficient fused implementation. We evaluate ESTRA at target densities of 5% and 1%. All sparse methods are applied from the first denoising step, without dense warmup, to assess how well they preserve generation over the entire sampling trajectory. All methods use 50 denoising steps. ESTRA uses 300 query clusters and 1,000 key clusters on both backbones.

Evaluation. We measure both fidelity to dense generation and the quality of the generated videos. PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) measure fidelity against dense outputs generated from the same prompts and initial noise. For video quality, we report four VBench dimensions (Huang et al., 2024): subject consistency, background consistency, imaging quality, and aesthetic quality. The evaluation covers their 251 distinct official GPT-enhanced prompts, with one video per prompt and a shared seed of 42 across methods. Fidelity metrics use all paired videos. Subject and background consistency use their designated 72- and 86-prompt subsets, while imaging and aesthetic quality share the remaining 93 prompts.

Efficiency. ESTRA uses FlashInfer (Ye et al., 2025) for selected-token attention and custom Triton (Tillet et al., 2019) kernels for selection and corrected cluster compensation. We report median end-to-end latency on NVIDIA H20, including all method overheads, VAE decoding, and video export, and speedup over dense generation. Density is the fraction of video query–key interactions computed exactly. We report target densities for ESTRA and measured densities for the baselines.

5.2 GENERATION QUALITY AND SPEED

At 5% density, ESTRA outperforms Sol-Attn and SVG-EAR in all three fidelity metrics on both backbones, despite their higher densities of 7.6–12.8% (Table 1). Relative to SVG-EAR, the PSNR gains are 3.10 dB on Wan and 1.11 dB on HunyuanVideo. ESTRA is faster than both baselines, with end-to-end speedups over dense generation of $3.26\times$ on Wan and $4.71\times$ on HunyuanVideo.

On VBench, ESTRA at 5% also exceeds both baselines in three of the four dimensions on each backbone. All four scores remain within one point of dense generation. In the Wan example in Figure 1, ESTRA retains the dense reference's full-body framing and placement of the polar bear, whereas Sol-Attn and SVG-EAR produce different compositions. More visual results can be seen in Appendix D.

Generation at 99% sparsity. At 1% density, ESTRA uses only one-fifth of the exact-interaction budget at 5%, reaching $3.55\times$ and $5.39\times$ end-to-end speedups on Wan and HunyuanVideo. To our knowledge, this is the first training-free sparse attention method to preserve high-quality video generation at 1% density without dense warmup. It still exceeds Sol-Attn in all three fidelity metrics on both backbones with less than one-twelfth as many exact interactions.

On HunyuanVideo, subject and background consistency exceed both baselines, while imaging and aesthetic quality remain within about one point of dense generation. Wan is more sensitive: imaging and aesthetic quality fall roughly five points below dense generation. Additional qualitative comparisons at approximately 1% density show that ESTRA retains clear subjects and coherent scenes while Sol-Attn and our cluster-level variant exhibit severe artifacts or blur (Appendix E).

Table 1: **Video quality, fidelity, and speed at 720p.** Fidelity metrics average 251 sparse–dense pairs per backbone; VBench scores use designated prompt subsets. Latency is the median H20 time per video, including VAE decoding and export. Bold marks the best sparse result per backbone. Subj., Bkgd., Imag., and Aesth. denote subject consistency, background consistency, imaging quality, and aesthetic quality.

	Efficiency			Fidelity to dense			VBench dimensions ↑			
Method	Density	Latency (s) ↓	Speedup	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	Subj.	Bkgd.	Imag.	Aesth.
Wan2.1-14B, *1280×720, 81 frames; no dense warmup*										
Dense	100%	6838	1.00×	ref.	ref.	ref.	91.78	94.42	66.98	66.02
Sol-Attn	12.8%	2139	3.20×	10.71	0.283	0.737	90.00	93.14	65.69	60.23
SVG-EAR	7.6%	2171	3.15×	11.57	0.342	0.657	89.80	92.13	**66.28**	63.79
ESTRA	5.0%	2097	3.26×	**14.67**	**0.489**	**0.467**	**91.06**	**93.59**	66.08	**65.93**
ESTRA	1.0%	**1928**	**3.55**×	12.25	0.377	0.668	89.97	92.23	61.87	60.74
HunyuanVideo-13B, *1280×720, 129 frames; no dense warmup*										
Dense	100%	6490	1.00×	ref.	ref.	ref.	93.13	95.25	67.78	66.32
Sol-Attn	12.5%	1448	4.48×	13.66	0.513	0.472	90.52	94.74	67.62	**67.13**
SVG-EAR	12.2%	1616	4.02×	17.89	0.666	0.302	93.98	95.49	67.14	65.93
ESTRA	5.0%	1378	4.71×	**19.00**	**0.700**	**0.266**	94.12	**95.67**	**67.90**	66.26
ESTRA	1.0%	**1203**	**5.39**×	16.58	0.620	0.378	**95.16**	95.64	66.75	65.20

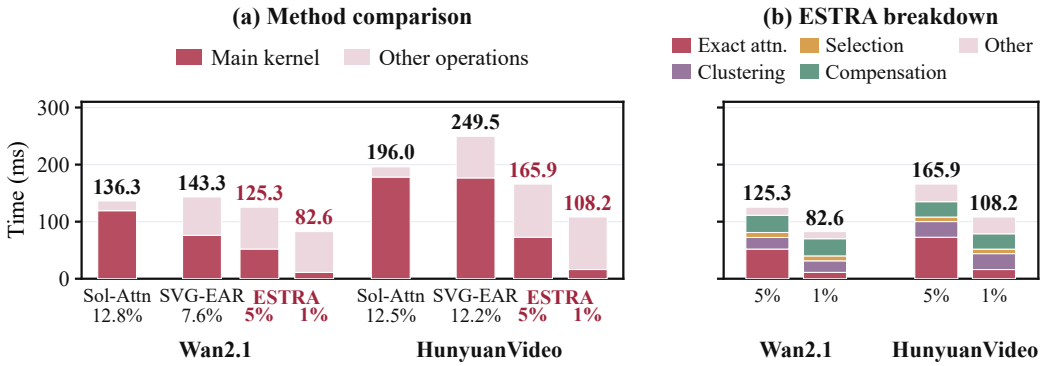

Figure 4: **Attention runtime on NVIDIA H20.** Panel (a) compares ESTRA with Sol-Attn and SVG-EAR on both backbones. Panel (b) breaks down ESTRA's runtime at 5% and 1% density.

Attention runtime. ESTRA incurs slightly higher overhead than SVG-EAR due to token-level scoring and VCM (Figure 4(a)). This additional computation allows ESTRA to achieve higher fidelity at a lower density (Table 1), so the savings in exact attention more than offset the extra cost. At 5% density, total attention time is 12.5% lower than SVG-EAR on Wan and 33.5% lower on HunyuanVideo, and is also lower than Sol-Attn on both backbones. Figure 4(b) shows that reducing density further to 1% mainly reduces selected-token attention time, while clustering, selection, and compensation costs remain nearly unchanged.

Table 2: **Fidelity at comparable attention budgets.** Results average 10 prompts per backbone at the densities shown. Bold marks the best score within each density group and backbone.

	Wan2.1-14B				**HunyuanVideo-13B**			
Method	Density	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	Density	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
Density: 5–6%								
Sol-Attn	5.0%	11.31	0.343	0.816	5.0%	12.29	0.453	0.545
SVG-EAR	5.9%	11.39	0.335	0.693	5.8%	18.41	0.693	0.282
ESTRA	5.0%	**14.64**	**0.507**	**0.455**	5.0%	**19.29**	**0.733**	**0.226**
Density: approximately 15%								
Sol-Attn	14.8%	10.55	0.292	0.761	15.0%	14.32	0.535	0.454
SVG-EAR	15.1%	12.11	0.359	0.623	15.1%	18.90	0.722	0.239
ESTRA	15.0%	**17.59**	**0.632**	**0.310**	15.0%	**20.54**	**0.764**	**0.197**

5.3 Fidelity across Attention Densities

To examine fidelity across attention budgets, we sweep densities from 1% to approximately 15% on ten prompts per backbone, using matched initial noise and dense reference videos. With these routing settings and prompts, SVG-EAR's minimum measured density is approximately 6%.

Figure 5 shows that ESTRA achieves a better fidelity–density trade-off on both backbones. At comparable densities, it outperforms Sol-Attn throughout the sweep and SVG-EAR over the shared density range in all three metrics. Its fidelity improves across all three metrics as density increases from 1% to 10%. The PSNR gains reach 5.61 dB on Wan and 4.12 dB on HunyuanVideo.

Table 2 compares all three methods within two density ranges, 5–6% and approximately 15%, using the density convention above. Even at 5% density, ESTRA outperforms SVG-EAR at 15.1% in all three metrics on both backbones, using roughly one-third as many exact interactions.

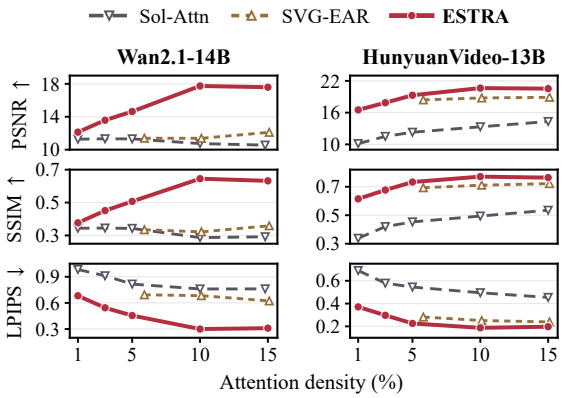

Figure 5: Fidelity to dense generation across attention densities. Results average 10 prompts per backbone.

5.4 Ablation of Selection and Compensation

The attention-output comparison in Figure 3 shows successive error reductions from VCM and token-level selection. We now evaluate whether these improvements increase fidelity over complete video generation. Table 3 follows the same three-stage ablation at a 5% target density, using the ten-prompt sets from the density study and the same dense reference videos. The starting variant ranks key clusters by their aggregate attention scores, selects whole clusters within a token budget, and uses count-weighted mean compensation.

Adding VCM to cluster-level selection improves PSNR by 0.27 dB on Wan and 0.44 dB on HunyuanVideo. Token-level selection then provides a further gain of 3.12 dB and 1.75 dB, respectively. Both stages also improve SSIM and LPIPS on both backbones.

Table 3: **End-to-end component ablation at a 5% target density.** Fidelity to dense generation is averaged over 10 prompts per backbone. Rows successively add VCM and token-level selection. Cluster-level densities are measured; bold marks the best score per backbone.

		Wan2.1-14B			HunyuanVideo-13B		
Variant	Density	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
Cluster selection	4.94%	11.25	0.325	0.698	17.10	0.654	0.336
+ VCM	4.94%	11.52	0.349	0.685	17.54	0.668	0.320
+ Token selection (**ESTRA**)	5.00%	**14.64**	**0.507**	**0.455**	**19.29**	**0.733**	**0.226**

6 Conclusion

We introduced ESTRA, a training-free method for accelerating video diffusion transformers. It uses query centroids to score individual keys efficiently, reserving exact attention for selected tokens and approximating the remaining contributions with cluster representatives. VCM adjusts the representative weights to further reduce approximation error at low densities. On Wan2.1-14B and HunyuanVideo-13B, ESTRA at just 5% attention density generates videos faster than Sol-Attn and SVG-EAR at higher densities. It also achieves higher fidelity to dense generation and better scores on most evaluated VBench dimensions. Even at 99% sparsity, ESTRA remains more faithful to dense generation than Sol-Attn on both models, while achieving end-to-end speedups of $3.55\times$ and $5.39\times$ over dense generation, respectively.

AI USE STATEMENT

Generative AI tools assisted with literature review, method development, implementation, experiment design and analysis, manuscript drafting and editing, and figure preparation. We did not use generative AI tools to generate synthetic data, clean datasets, or perform qualitative data analysis, and translation is not applicable to this work. The authors directed the research, reviewed and revised the AI-assisted material, and take responsibility for the content of the manuscript.

ETHICS STATEMENT

This work accelerates inference of publicly released video generation models without training and introduces no new models or datasets; all experiments use the public VBench prompts and involve no human subjects or personal data. Because ESTRA changes only the cost of generation and not what the underlying models can generate, it neither adds to nor mitigates their potential for misuse. Reducing inference cost also lowers the energy consumption of video generation.

REPRODUCIBILITY STATEMENT

The method is fully specified by the equations in Section 4, and Appendix C gives the joint attention computation together with the production configuration, including clustering schedules, threshold estimation, buffer capacity, numerical precision, the VCM cap, and the baseline settings. Section 5 states the backbones, resolutions, samplers, denoising steps, prompt sets, seed, evaluation metrics, and timing hardware; the models, baseline implementations, and VBench prompts used are all publicly available. The protocols of the dense-warmup comparison and the offline attention-error analysis are described in Appendices A and B.

REFERENCES

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, Yuanzhen Li, Michael Rubinstein, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, 2024. doi: 10.1145/3680528.3687614. Article 94.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359, 2022. doi: 10.52202/068431-1189.

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

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. *arXiv preprint arXiv:2607.24027*, 2026a. URL https://arxiv.org/abs/2607.24027.

Haopeng Li, Shitong Shao, Wenliang Zhong, Zikai Zhou, Lichen Bai, Hui Xiong, and Zeke Xie. PISA: Piecewise sparse attention is wiser for efficient diffusion transformers. *arXiv preprint arXiv:2602.01077*, 2026b. URL https://arxiv.org/abs/2602.01077.

Enhuai Liu, Yunke Wang, Changming Sun, and Chang Xu. D2Cache: Second-order delta caching for higher video diffusion acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43589–43599, 2026a.

Enhuai Liu, Yunke Wang, Yutong Wang, Changming Sun, and Chang Xu. LoSA: Near-lossless sparse attention for training-free video diffusion acceleration. *arXiv preprint arXiv:2608.12032*, 2026b. URL https://arxiv.org/abs/2608.12032.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7353–7363, 2025.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5775–5787, 2022. doi: 10.52202/068431-0418.

Zhengyao Lv, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K. Wong. FasterCache: Training-free video diffusion model acceleration with high quality. In *International Conference on Learning Representations*, pp. 33132–33156, 2025. URL https://openreview.net/forum?id=W49UjcpGxx.

Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=ntGPYNUF3t.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, et al. Movie Gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. URL https://arxiv.org/abs/2410.13720.

Austin Silveria, Soham V. Govande, and Daniel Y. Fu. Chipmunk: Training-free acceleration of diffusion transformers with dynamic column-sparse deltas. *arXiv preprint arXiv:2506.03275*, 2025. URL https://arxiv.org/abs/2506.03275.

Philippe Tillet, H. T. Kung, and David Cox. Triton: An intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pp. 10–19, 2019. doi: 10.1145/3315508.3329973.

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

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

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. URL https://proceedings.mlr.press/v267/xi25c.html.

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, 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, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, pp. 83048–83077, 2025b. URL https://openreview.net/forum?id=LQzN6TRFg9.

Zihao Ye, Lequn Chen, Ruihang Lai, Wuwei Lin, Yineng Zhang, Stephanie Wang, Tianqi Chen, Baris Kasikci, Vinod Grover, Arvind Krishnamurthy, and Luis Ceze. FlashInfer: Efficient and customizable attention engine for LLM inference serving. In *Proceedings of Machine Learning and Systems*, volume 7, 2025.

Jintao Zhang, Jia Wei, Pengle Zhang, Jun Zhu, and Jianfei Chen. SageAttention: Accurate 8-bit attention for plug-and-play inference acceleration. In *International Conference on Learning Representations*, pp. 71566–71585, 2025a. URL https://openreview.net/forum?id=OL44KtasKc.

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, 2025b. URL https://proceedings.mlr.press/v267/zhang25ch.html.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74714–74731, 2025c. URL https://proceedings.mlr.press/v267/zhang25m.html.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

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. In *Computer Vision – ECCV 2026*, volume 17050 of *Lecture Notes in Computer Science*, pp. 167–185, 2026. doi: 10.1007/978-3-032-37359-5_10.

APPENDIX

A COMPARISON WITH DENSE WARMUP

The no-warmup setting in our main experiments directly evaluates how well each sparse method preserves generation throughout the sampling trajectory. Applying the same schedule to all methods also controls for the amount of dense computation. Dense warmup can improve fidelity by computing the initial steps exactly, at the cost of additional computation. We therefore also compare the methods with a shared dense warmup.

The evaluation uses the same ten prompts per backbone, initial noise, and dense reference videos as the density study. All methods use SVG-EAR's dense attention path for the first 10 of 50 denoising steps and their respective sparse implementations for the remaining 40. Density controls and other inference settings follow Table 1. We report mean end-to-end latency on NVIDIA H20, including VAE decoding and video export, and compute speedups relative to the dense reference times in Table 1.

Table 4: **Comparison with a shared dense warmup.** All methods use 10 dense steps followed by 40 sparse steps. Fidelity metrics and end-to-end latency are averaged over ten prompts per backbone. Density refers to the sparse phase; including warmup, the trajectory-averaged density is approximately $20\% + 0.8\rho$. Speedups use the dense reference times in Table 1. Bold marks the best result per backbone.

Method	Sparse-phase density ρ	Latency (s) ↓	Speedup	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
Wan2.1-14B						
Sol-Attn	12.8%	3086	2.22×	18.88	0.587	0.296
SVG-EAR	7.1%	3077	2.22×	21.17	0.727	0.200
ESTRA	5.0%	3020	2.26×	**21.52**	**0.746**	**0.191**
ESTRA	1.0%	**2884**	**2.37**×	20.69	0.708	0.245
HunyuanVideo-13B						
Sol-Attn	12.5%	1955	3.32×	22.89	0.805	0.150
SVG-EAR	12.1%	2088	3.11×	26.01	0.873	0.079
ESTRA	5.0%	1891	3.43×	**27.66**	**0.898**	**0.066**
ESTRA	1.0%	**1752**	**3.70**×	24.41	0.831	0.134

Table 4 shows that ESTRA at 5% sparse-phase density still outperforms both baselines in all three fidelity metrics on both backbones, while also generating videos faster. Relative to the matched no-warmup runs, its PSNR increases from 14.64 to 21.52 dB on Wan and from 19.29 to 27.66 dB on HunyuanVideo. These warmup settings retain end-to-end speedups of $2.26\times$ and $3.43\times$ over the dense reference, respectively. The 1% setting increases these speedups to $2.37\times$ and $3.70\times$ while remaining more faithful to dense generation than Sol-Attn in all three metrics.

B Additional Offline Analysis

Evaluation protocol. The offline study uses ten prompts per backbone on an NVIDIA H20, with identical QKV inputs from step 10 of dense generation. It covers every attention layer (40 for Wan and 60 for HunyuanVideo), head, and video query row. Each variant uses its production attention implementation; inputs and clustering state are restored before each evaluation. Dense outputs propagate to subsequent layers, so sparse approximations do not alter the inputs being compared. The reference is exact attention computed in FP32, including valid text keys and excluding padding. For a reference output vector $o^\star$ and its approximation $\widetilde{o}$, we measure

$$e(\widetilde{o}, o^\star) = \frac{\|\widetilde{o} - o^\star\|_2}{\|o^\star\|_2}, \tag{10}$$

and average these relative output errors over layers and prompts.

B.1 Complete Component Ablation

The cluster-level variant uses the same query-centroid scores a_{gj} as ESTRA (Eq. (9)), aggregated into a score for each key cluster:

$$b_{g,c} = \log \sum_{j \in \mathcal{C}_c} \exp(a_{gj}). \tag{11}$$

Clusters are considered in descending score order. The first cluster is retained, and subsequent clusters are admitted only within the remaining token budget, without splitting their members. The realized densities are 0.9%, 2.9%, 4.9%, 9.9%, and 14.9% for target budgets of 1%, 3%, 5%, 10%, and 15%, respectively. Token-level settings are labeled by their target densities.

Table 5 reports both selection granularities with and without VCM, including the token-level variant without VCM that is not plotted in Figure 3. For either granularity, toggling VCM leaves the selected tokens, K/V means, and remaining counts unchanged. Token-level selection lowers error with or without VCM across the full sweep, confirming that its benefit does not depend on the correction. VCM reduces error for both selection granularities at 1–5% density. At higher densities, it continues to help whole-cluster selection but can increase error with token-level selection. Thus, the correction's benefit depends on both the density and the selected tokens.

Table 5: **Complete component ablation.** Attention-output errors average 10 prompts per backbone. Token selection uses target density labels; whole-cluster selection uses measured densities. Bold marks the lower error within each VCM pair.

	Cluster selection			Token selection	
Budget	Density	w/o VCM	w/ VCM	w/o VCM	w/ VCM
Wan2.1-14B					
1%	0.9%	0.3719	**0.3484**	0.2001	**0.1775**
3%	2.9%	0.2362	**0.2205**	0.1172	**0.1043**
5%	4.9%	0.1854	**0.1720**	0.0873	**0.0769**
10%	9.9%	0.1258	**0.1149**	0.0540	**0.0522**
15%	14.9%	0.0967	**0.0874**	**0.0391**	0.0427
HunyuanVideo-13B					
1%	0.9%	0.2223	**0.1779**	0.1423	**0.1136**
3%	2.9%	0.1467	**0.1141**	0.0820	**0.0682**
5%	4.9%	0.1156	**0.0889**	0.0603	**0.0539**
10%	9.9%	0.0795	**0.0609**	**0.0368**	0.0395
15%	14.9%	0.0615	**0.0473**	**0.0263**	0.0328

B.2 Comparison of Cluster Selection Rules

Figure 6 compares the mass-ranked cluster selection above with SVG-EAR's official efficiency-based routing, each with and without VCM. The two selectors use the same clusters and production block-sparse attention kernel. The SVG-EAR operating points are listed in Table 6.

At realized densities near 10% and 15%, the two selectors produce comparable errors without VCM. VCM reduces error for both, indicating that its benefit also extends to SVG-EAR's selected clusters. The simpler mass ranking therefore provides an effective cluster-level reference for evaluating token-level selection.

The lowest realized density differs substantially between the selectors. For requested budgets of 1–5%, the evaluated SVG-EAR routing policy yields densities of 5.5–5.7%, whereas mass-ranked selection reaches 0.9% at a 1% budget. The approximately 6% minimum observed for SVG-EAR is therefore a property of its routing and budget policy on these inputs, rather than an inherent limit of whole-cluster selection. All curves use realized densities to keep these differences visible.

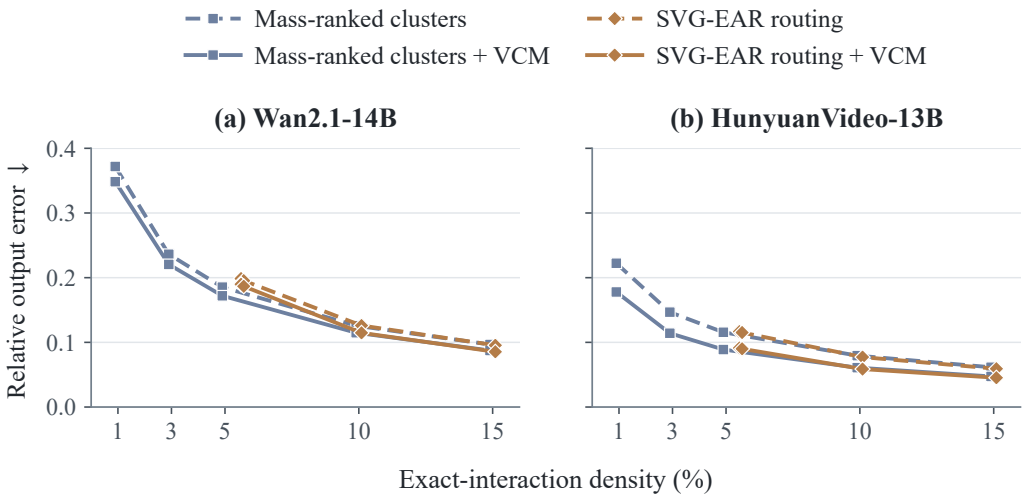

Figure 6: **Cluster selection and VCM.** Mass-ranked selection and SVG-EAR routing are each evaluated with and without VCM on identical inputs. Each VCM pair retains the same selected clusters. Curves use measured densities and cover all requested budgets, including SVG-EAR's overlapping low-budget points. Errors average 10 prompts per backbone.

Table 6: **SVG-EAR routing with and without VCM.** The two variants use the same selected clusters. Budget is the requested density; density is measured. Errors average 10 prompts per backbone. All requested budgets are retained, including those with the same rounded measured density.

	Wan2.1-14B			**HunyuanVideo-13B**		
Budget	Density	w/o VCM	w/ VCM	Density	w/o VCM	w/ VCM
1%	5.6%	0.1987	0.1898	5.5%	0.1166	0.0913
3%	5.6%	0.1986	0.1898	5.5%	0.1166	0.0913
5%	5.7%	0.1954	0.1867	5.6%	0.1152	0.0900
10%	10.1%	0.1260	0.1147	10.1%	0.0772	0.0587
15%	15.1%	0.0955	0.0854	15.1%	0.0592	0.0453

B.3 Comparison with Sparse Attention Baselines

Figure 7 compares full ESTRA with SVG-EAR and Sol-Attn under the same offline protocol. SVG-EAR's complete results, including the additional VCM variant, appear in Table 6; Table 7 reports Sol-Attn's thresholds and measured densities. At 5% density, ESTRA has lower error on both backbones than SVG-EAR at 15.1% and the highest-density Sol-Attn settings evaluated (20.7% on Wan and 17.8% on HunyuanVideo). Even after adding VCM to SVG-EAR, ESTRA at 5% remains more accurate than SVG-EAR at 10.1% on both backbones. At 1%, ESTRA also has lower error than unmodified SVG-EAR near its minimum realized density of 5.5–5.6%.

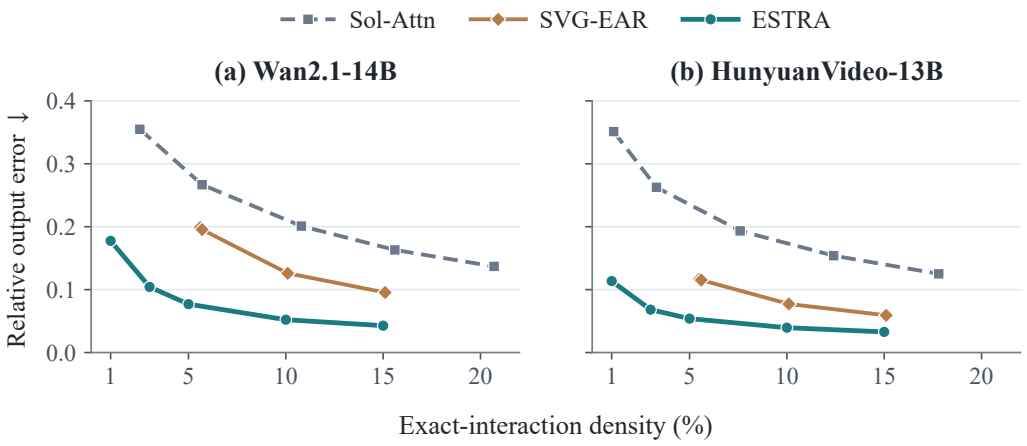

Figure 7: **Attention-output error across methods.** Production implementations are evaluated on identical QKV inputs from step 10 of dense generation. Errors cover all layers, heads, and video query rows, averaged over 10 prompts per backbone. Baselines use measured densities; ESTRA uses target densities and VCM at every point.

Table 7: **Sol-Attn operating points for the offline comparison.** The official kernel is evaluated at five thresholds τ; density is measured. Errors average the same 10 prompts per backbone.

	Wan2.1-14B		**HunyuanVideo-13B**	
τ	Density	Error ↓	Density	Error ↓
1.3	20.7%	0.1369	17.8%	0.1251
1.7	15.6%	0.1631	12.4%	0.1540
2.2	10.8%	0.2010	7.6%	0.1934
3.0	5.7%	0.2668	3.3%	0.2626
4.0	2.5%	0.3549	1.1%	0.3512

C Attention Evaluation and Cost

Production configuration. Clustering is independent for each layer and attention head, with 50 iterations from randomly sampled token centroids at the first step and two iterations per subsequent step, initialized from the previous centroids. Token selection is recomputed at every attention call. For Wan, key assignments use a 16-dimensional projection estimated from 4,096 sampled keys and refreshed every ten steps; centroid updates and attention remain full-dimensional. Query clustering and all HunyuanVideo clustering use the full dimension. Models and attention kernels run in BF16, with FP32 statistics; Wan's VAE runs in FP32. The VCM cap $\delta_{\max}$ is 1 for HunyuanVideo and for Wan at target densities of 5% or above, and 2 for Wan at 3% or below.

For the main comparison, Sol-Attn uses $\tau = 1.7$ on Wan and 1.6 on HunyuanVideo; SVG-EAR uses requested budgets of 7% and 12%, respectively, with its minimum-cluster-ratio constraint disabled. The sampling shifts are 5 for Wan's second-order UniPC solver and 7 for HunyuanVideo's FlowMatch-Euler solver.

Each original query $i \in \mathcal{G}_g$ forms a separate row in Figure 2(d), attending jointly to selected tokens and calibrated representatives:

$$\tilde{o}_i = \frac{\sum_{j \in T_g} e^{s_{ij}} v_j + \sum_{c:\, r_{g,c} > 0} e^{\ell_{i,c}} \bar{v}_c}{\sum_{j \in T_g} e^{s_{ij}} + \sum_{c:\, r_{g,c} > 0} e^{\ell_{i,c}}}. \tag{12}$$

The log-count and VCM biases apply only to the representative columns and are shared across query rows within each cluster. The exact and approximate contributions share a normalizer, so calibration adjusts their relative weights. Separate kernels evaluate the selected tokens and cluster representatives, merging their contributions with online softmax without materializing the combined attention matrix.

For joint video–text attention, selection thresholds are estimated from video keys only. Valid text keys are always retained for video queries. Padding keys remain masked, and text-query outputs are computed densely.

We estimate selection thresholds from 2,048 uniformly strided keys, with a buffer capacity of 1.25 times the target token budget per query cluster. Keys are sampled from the reordered sequence; the threshold τ_g is the cutoff for the highest-scoring ρ fraction of sampled keys. We scan all keys and store indices satisfying $a_{gj} \geq \tau_g$ in the buffer. If more keys pass the threshold than the buffer can hold, only indices written within capacity are retained. Production logs show selected-token counts within 1% of the 5% target budget and 95–97% of the 1% target budget (approximately 0.96% density). The realized exact-interaction density is $\hat{\rho} = \sum_g |\mathcal{G}_g|\,|T_g|/(N_q N_k)$, which can differ from ρ because of threshold estimation and capacity limits. For the video–video densities reported in our experiments, these counts are restricted to video tokens. Counting a multiply-add as two FLOPs, the leading matrix products cost

$$\underbrace{2C_q N_k d}_{\text{token scoring}} + \underbrace{4\hat{\rho} N_q N_k d}_{\text{selected-token attention}} + \underbrace{4 N_q C_k d}_{\text{representative attention}}. \tag{13}$$

For fixed cluster counts, scoring and representative attention scale linearly with sequence length. For the 81-frame Wan videos at 720p, $N_q = 75{,}600$; with $C_q = 300$, full-key scoring costs about 0.4% of dense QK arithmetic. Clustering, threshold estimation, index preparation, and backbone-specific operations add overhead and are included in end-to-end timing.

D Additional Qualitative Comparisons

Figures 8–13 show all 12 examples in our qualitative set: seven from Wan2.1-14B and five from HunyuanVideo-13B. Each comparison uses the same prompt and initial noise across methods. We display full frames at 0.5, 2.0, 3.5, and 5.0 seconds, using identical timestamps for every method and example. Density and speedup labels follow the main-table configurations.

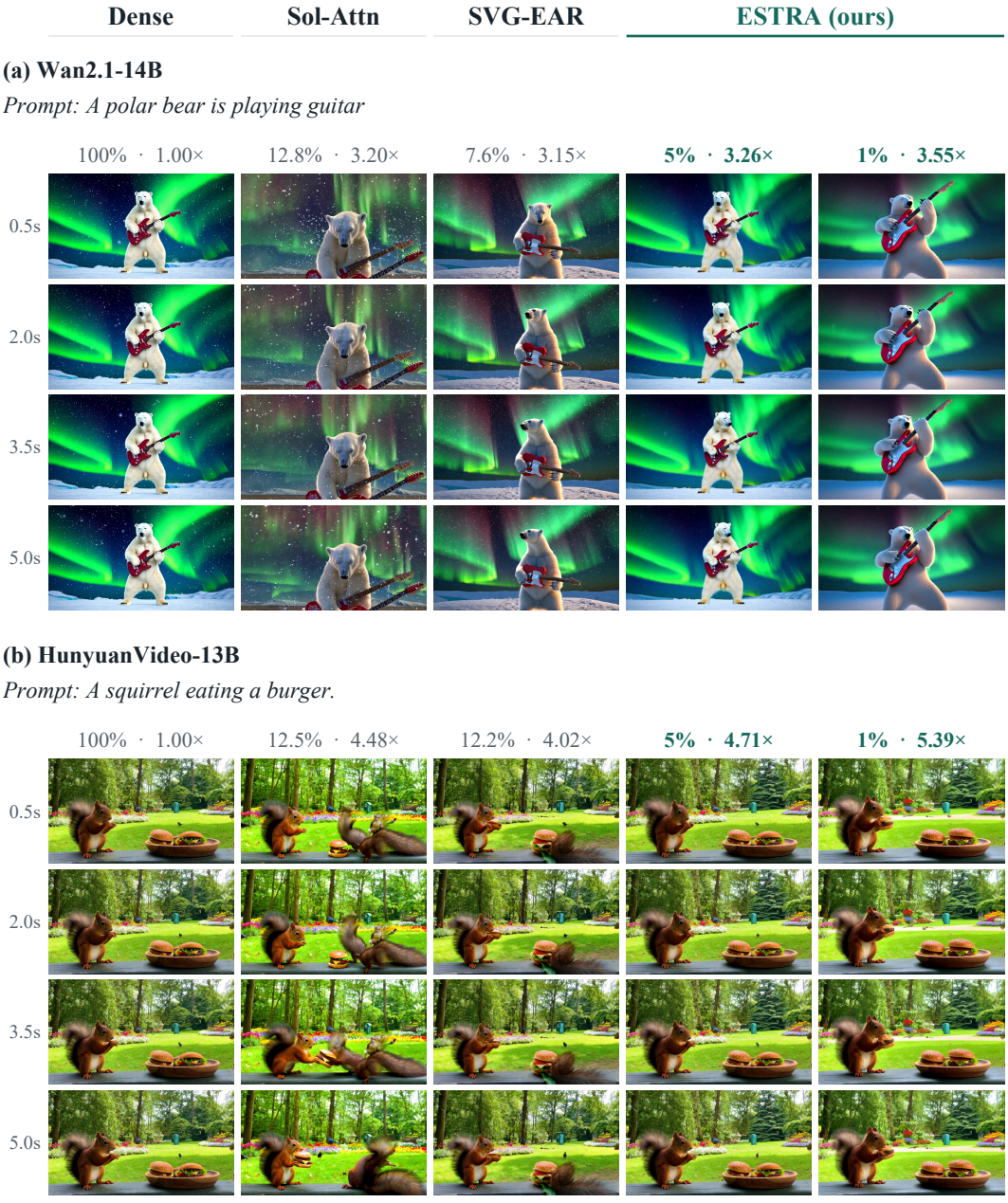

Figure 8: **Additional qualitative comparisons.** Each example shows the five main operating points, with full frames aligned at the indicated times.

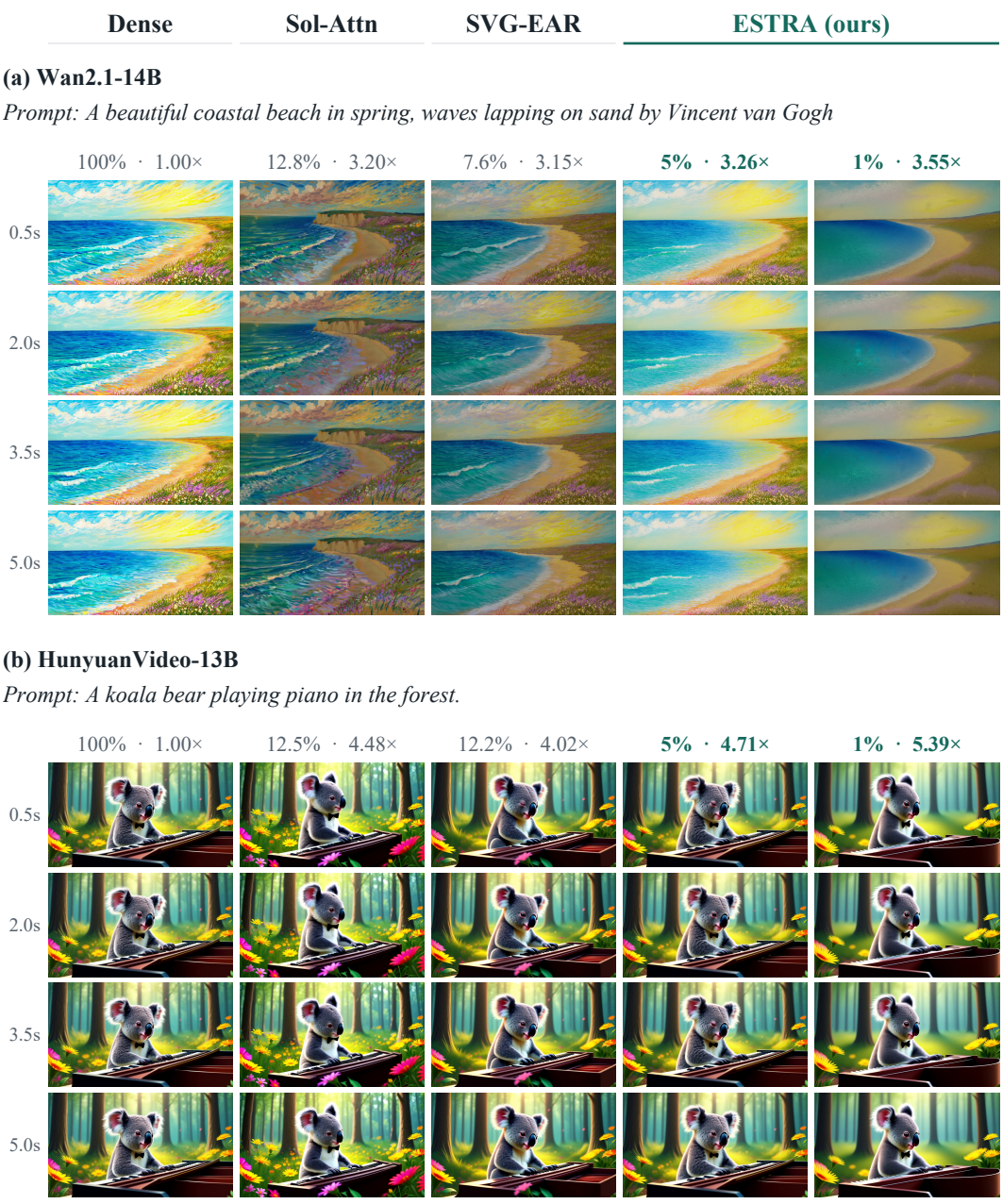

Figure 9: **Additional qualitative comparisons.** Each example shows the five main operating points, with full frames aligned at the indicated times.

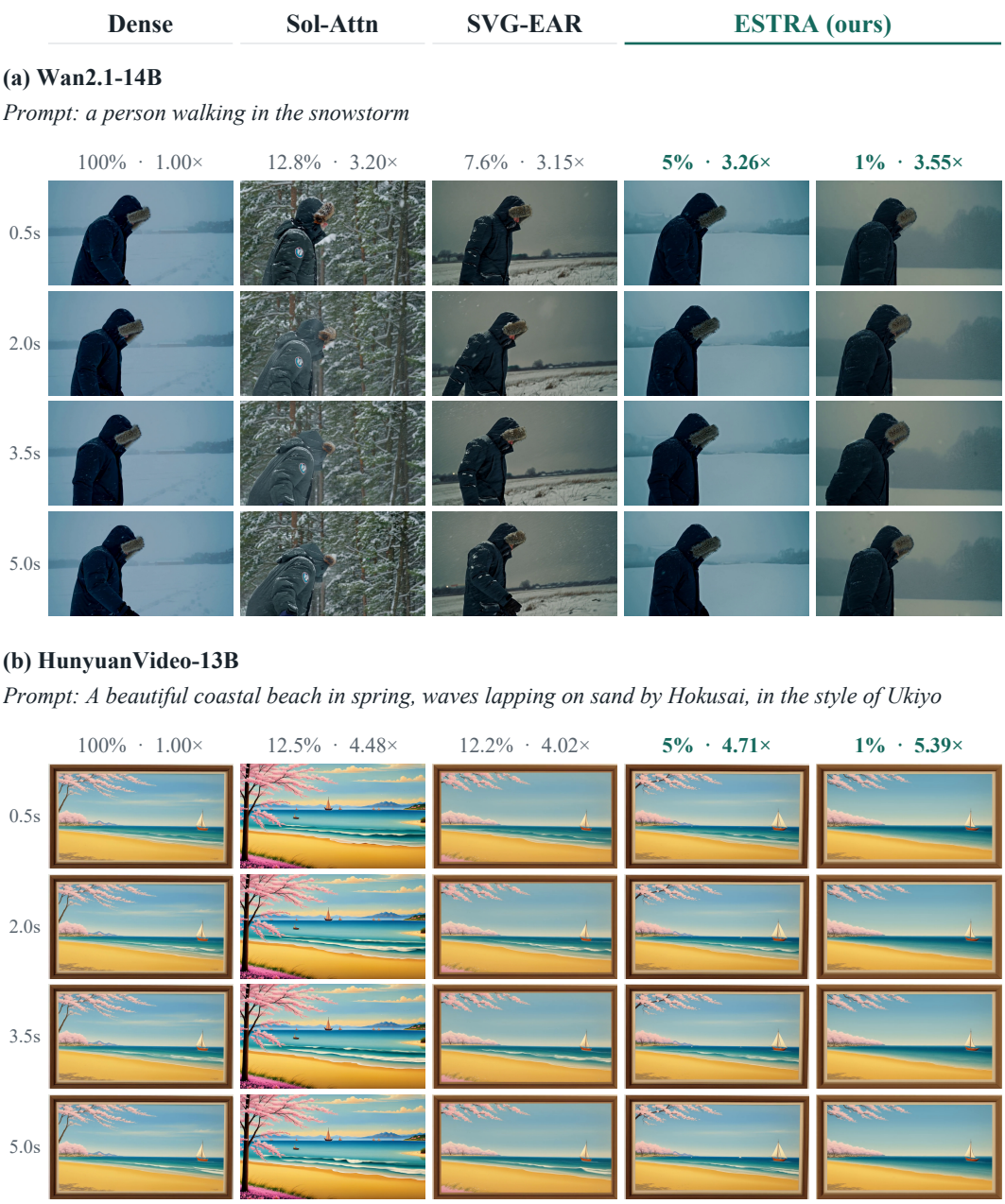

Figure 10: **Additional qualitative comparisons.** Each example shows the five main operating points, with full frames aligned at the indicated times.



Figure 11: **Additional qualitative comparisons.** Each example shows the five main operating points, with full frames aligned at the indicated times.

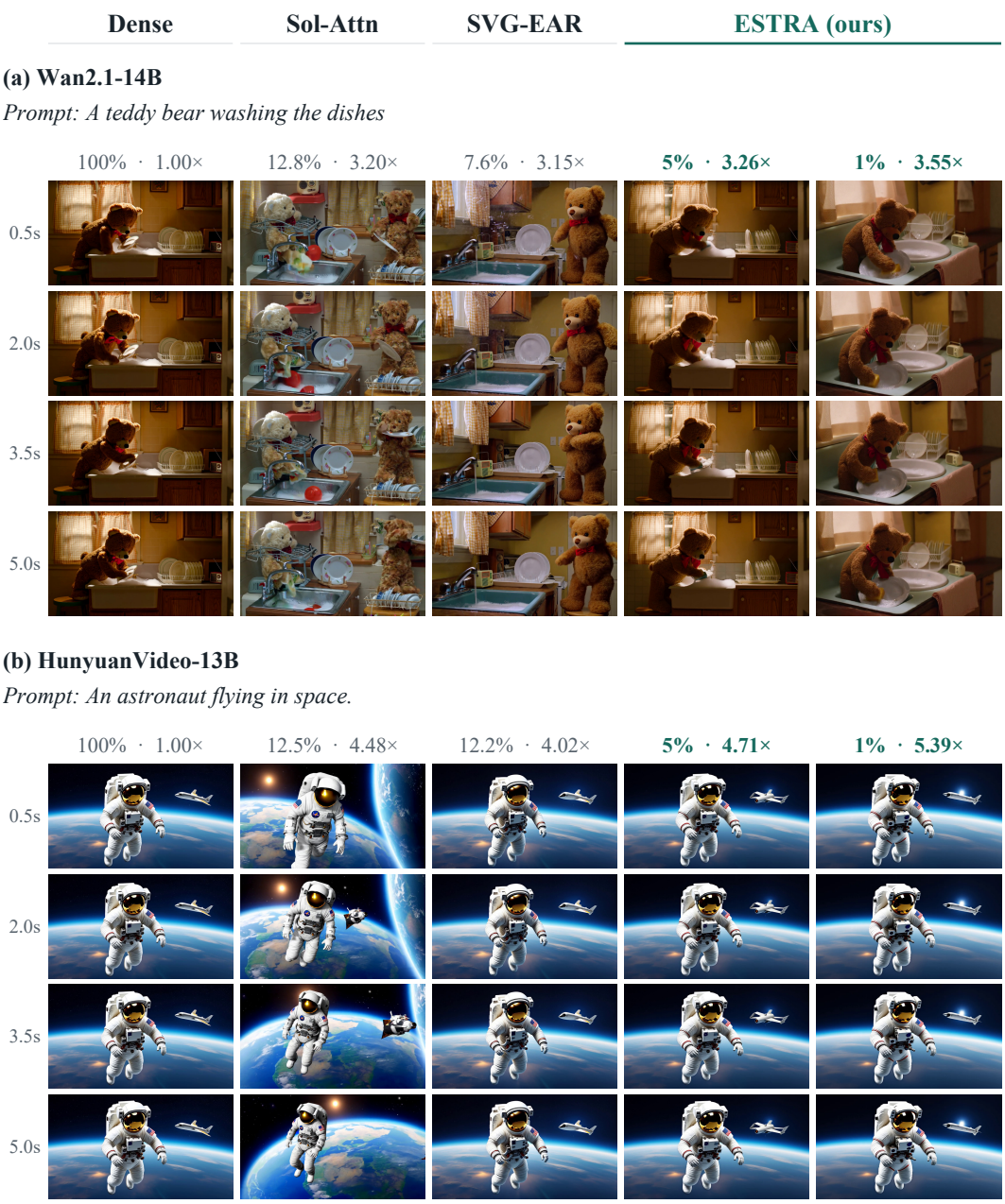

Figure 12: **Additional qualitative comparisons.** Each example shows the five main operating points, with full frames aligned at the indicated times.

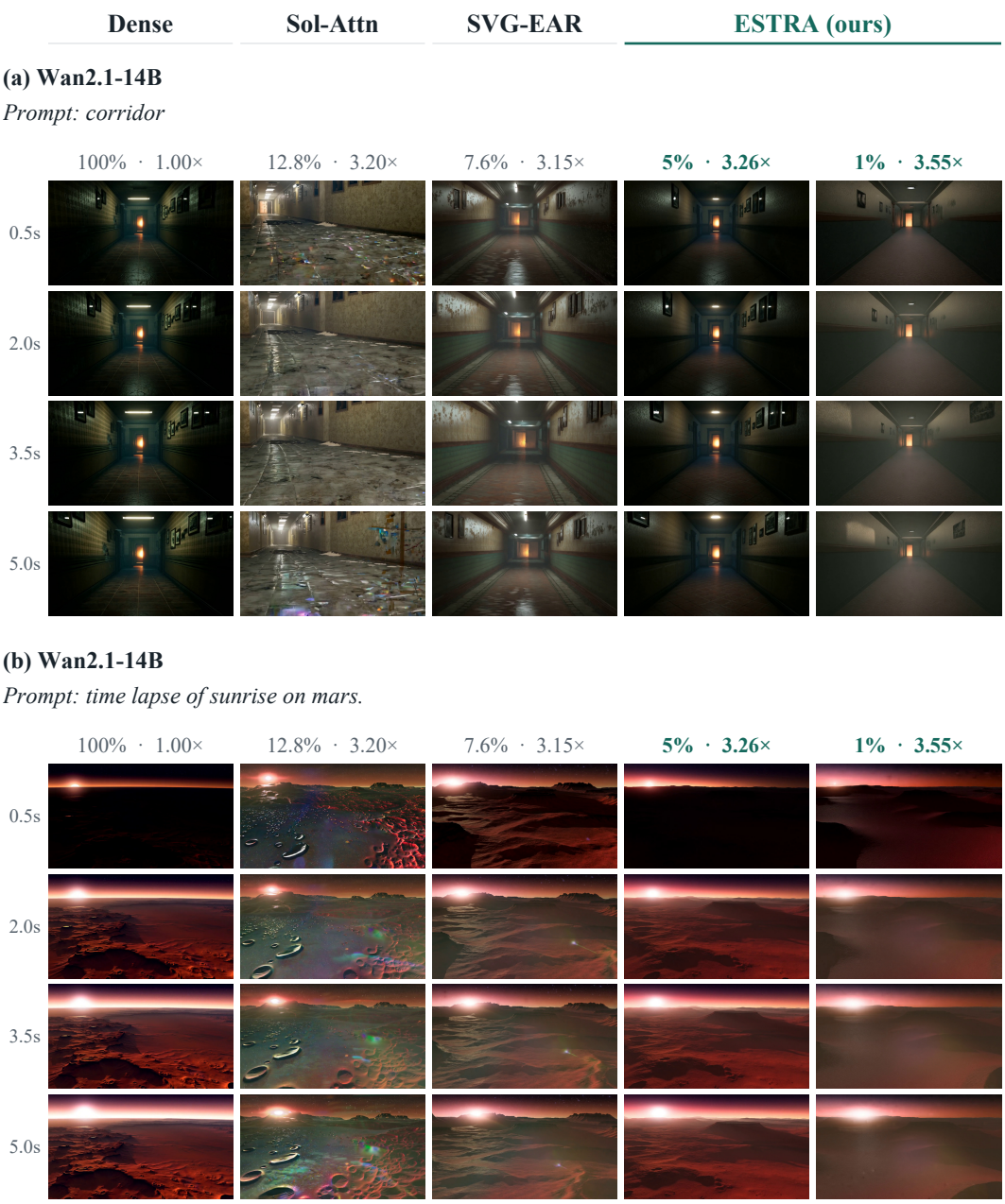

Figure 13: **Additional qualitative comparisons.** Each example shows the five main operating points, with full frames aligned at the indicated times.

E Qualitative Comparison at Extreme Sparsity

Figures 14 and 15 compare Sol-Attn, the cluster-level variant, and ESTRA at measured densities near 1%, alongside dense references. The four examples use the same prompt and initial noise across methods. The cluster-level variant uses whole-cluster selection and mean-based compensation without VCM, corresponding to the first stage of the component ablation in Section 5.4.



Figure 14: **Generation at approximately 1% density on Wan2.1-14B.** ESTRA preserves clear subjects and coherent scenes, whereas Sol-Attn exhibits severe block artifacts and the cluster-level variant loses substantial visual detail. Rows show aligned frames at the indicated times.

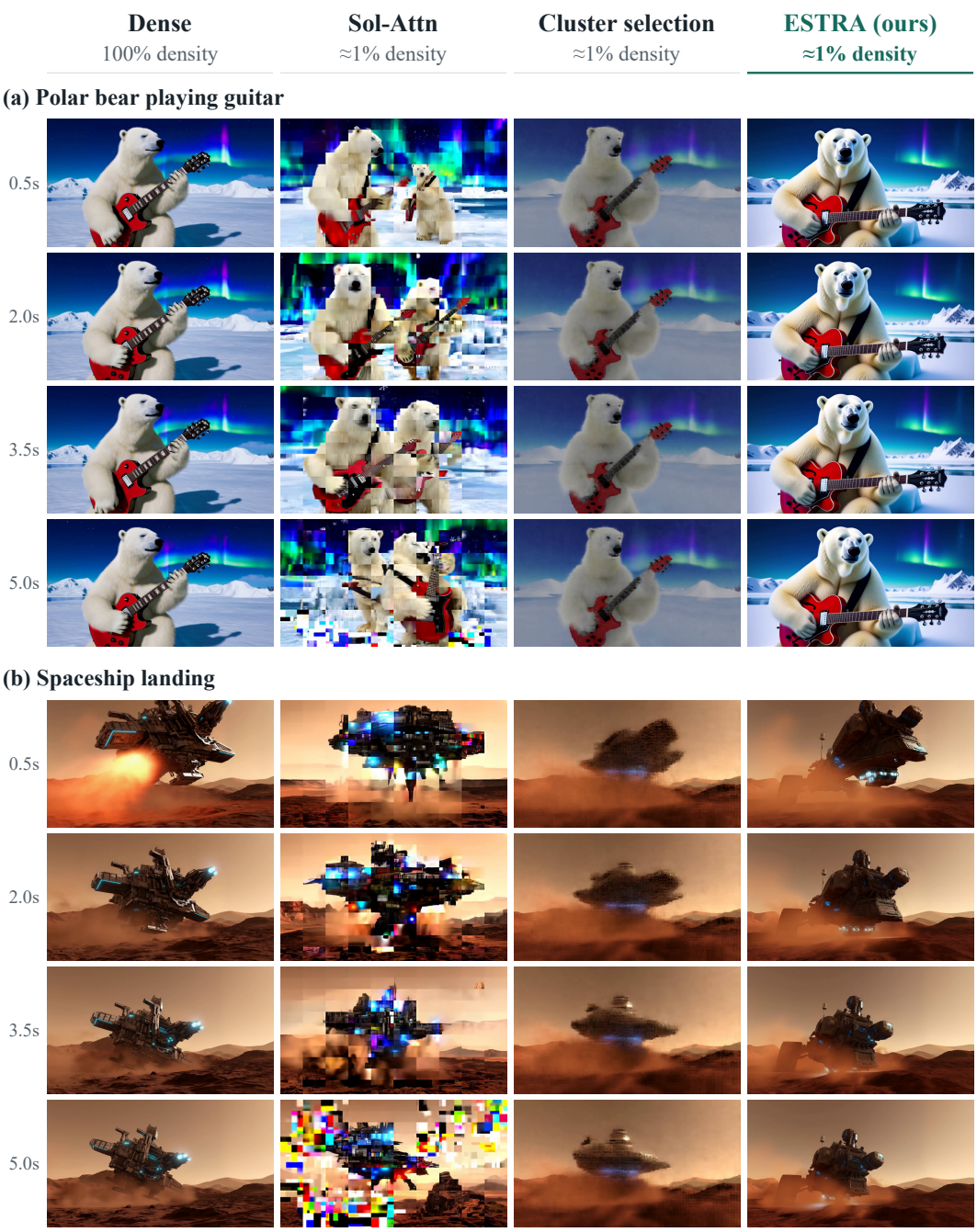

Figure 15: **Generation at approximately 1% density on HunyuanVideo-13B.** ESTRA retains detailed subjects and coherent scenes, while Sol-Attn and the cluster-level variant produce block artifacts and substantial blur, respectively. Rows show aligned frames at the indicated times.