RoLA: Rotary-Positioned Low-Rank Linear Attention for Efficient Diffusion Transformers

Anonymous authors
Paper under double-blind review

Abstract

Diffusion Transformers (DiTs) achieve strong video generation quality, but their dense spatiotemporal self-attention incurs quadratic complexity with sequence length, becoming the primary inference bottleneck as video resolution and frame count increase. Sparse low-rank hybrids address this by combining a local sparse branch with a global compressed branch. In video DiTs equipped with 3D Rotary Position Embeddings (RoPE), the global branch faces a structural compatibility issue: when RoPE is applied before a nonlinear feature map, the rotation and nonlinearity generally do not commute, making it difficult to keep a query-independent linear summary while preserving relative rotary geometry. Existing work often sidesteps this issue by replacing genuine cross-token global aggregation with coordinate-conditioned surrogates or learnable absolute positional modules. These compromises can be effective, but they approximate relative decay from absolute coordinates and introduce extra positional parameters. We propose **RoLA**, a rotary-positioned low-rank linear-attention branch that keeps genuine cross-token aggregation while remaining compatible with a reusable linear summary. The design applies RoPE *outside* the nonlinear low-rank feature map and reuses a truncated subset of the pre-trained rotary schedule matched to the low-rank bottleneck. This yields a linear-time low-rank global branch with relative positional behavior by design and no additional positional parameters; the full sparse–low-rank module still includes the fixed-sparsity sparse branch. Experiments on open-source video DiTs show that the resulting method remains competitive in generation quality at 90% sparsity while achieving 2.63× end-to-end inference speedup over dense attention on Wan2.1-14B (720p, 81 frames, measured on an NVIDIA H100 GPU).



Visual comparison of video generation quality across different attention methods.

1 Introduction

Diffusion Transformers (DiTs) (Peebles & Xie, 2023) have become a dominant backbone for high-fidelity video generation (Wan et al., 2025; Kong et al., 2024; Yang et al., 2025b; HaCohen et al., 2024). However, their dense spatiotemporal self-attention scales quadratically with sequence length,

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.

making inference increasingly expensive as video resolution and duration grow. Sparse attention methods (Xi et al., 2025; Yang et al., 2025a; Zhang et al., 2025b; Wu et al., 2025) reduce this cost by restricting each token to a local neighborhood. While effective at moderate sparsity, aggressive sparsification often removes the global interactions required for semantic consistency, motion coherence, and stable scene layout.

Sparse–linear hybrids address this limitation by augmenting a sparse local branch with a cheap global branch (Zhang et al., 2025a; 2026; Fang et al., 2026). In video DiTs, however, this design can encounter a structural compatibility issue. On the one hand, 3D Rotary Position Embeddings (RoPE) encode relative geometry through position-dependent rotations. On the other hand, linear attention relies on a query-independent global summary that can be reused across tokens. If RoPE is pushed through a nonlinear feature map in the usual way, the rotation and nonlinearity generally do not commute, so the resulting summary can become entangled with absolute positions. We refer to this tension as the **RoPE Dilemma**.

One practical workaround is to replace genuine cross-token aggregation in the global branch with a low-rank Fourier compensator built from coordinate-conditioned surrogates and learnable absolute positional modules (Liu et al., 2026b). This works, but it leaves two limitations. First, relative decay is approximated from absolute coordinates rather than represented by the global branch itself. Second, the design gives up the reusable linear-attention view of global context and introduces extra positional parameters.

This naturally raises the question: *can we build a genuine linear global branch that remains compatible with 3D relative-position structure?* We answer this question by rotating the already-compressed low-rank features *after* the nonlinearity. As a result, the branch keeps a reusable global summary while its pairwise interactions remain functions of relative offsets. We further reuse a truncated subset of the pre-trained rotary schedule that matches the low-rank bottleneck, avoiding any new positional module. Together, these two choices turn the compressed branch from an absolute-coordinate surrogate into a relative-position-aware linear branch, while preserving linear-time reuse and a hardware-friendly implementation. Although this outside-activation ordering is simple in isolation, our contribution is not the ordering as an isolated trick, but making it satisfy relative-position awareness, a reusable linear-time summary, sparse-engine compatibility, and zero additional positional parameters *simultaneously* in a pre-trained 3D-RoPE video DiT, together with the mechanistic and empirical evidence that it works. We refer to the resulting method as **RoLA**. Figure 1 summarizes the target operating regime, where RoLA reduces attention cost and end-to-end latency while retaining a genuine global branch; these deployment-oriented results use a separate RTX 5090 profiling setup and are not directly comparable to the H100-based main benchmark table.

Contributions. Our contributions are as follows:

- **Structural compatibility issue.** In 3D-RoPE video DiTs, a genuine low-rank global branch must preserve relative-position geometry while remaining reusable as a query-independent linear summary. These two requirements conflict under the usual nonlinear linear-attention formulation.
- **Compatible rotary low-rank branch.** The proposed branch applies RoPE after the nonlinear low-rank feature map and reuses a rank-truncated subset of the pre-trained rotary schedule. This preserves relative-position structure at the interaction level while keeping a reusable linear-time global summary, without introducing new positional parameters.
- **Mechanistic and empirical validation.** Mechanism-level experiments on real features from mainstream open-source video models verify the branch's relative-position behavior, and generation experiments across representative backbones demonstrate favorable quality–efficiency trade-offs under high sparsity and full-stack inference.

Related Work is provided in Appendix A.

2 PRELIMINARY

Notation. A video is flattened into L spatiotemporal tokens indexed by $p = (t, x, y) \in [T] \times [H_s] \times [W_s]$, with $L = TH_sW_s$. Each head projects the input to $Q, K, V \in \mathbb{R}^{L \times d_h}$, with per-token vectors $q_p, k_p, v_p \in \mathbb{R}^{d_h}$. We use p and q as token-position indices; q_p and k_q denote the query and key vectors at those positions. Full attention computes $O = \mathrm{softmax}(QK^\top/\sqrt{d_h})V$, whose $\mathcal{O}(L^2)$ cost is the central bottleneck for ultra-long sequences ($L \gtrsim 10^4$) induced by high-resolution video.

3D RoPE. 3D RoPE partitions the head dimension across the temporal and spatial axes, with $d_h = d_t + d_x + d_y$, and applies a block-diagonal orthogonal rotation $\mathbf{R}(p) \in \mathbb{R}^{d_h \times d_h}$. Its orthogonality gives the relative-position identity

$$s_{p,q} = \frac{1}{\sqrt{d_h}} \langle \mathbf{R}(p) q_p, \mathbf{R}(q) k_q \rangle = \frac{1}{\sqrt{d_h}} \langle q_p, \mathbf{R}(q-p) k_q \rangle. \tag{1}$$

RoPE Dilemma. Linear attention (Katharopoulos et al., 2020) relies on a query-independent global summary that can be reused across queries, whereas RoPE encodes relative geometry through position-dependent rotations. For a feature map ϕ, an inside-RoPE linear branch would form a key-side summary such as $\sum_q \phi(\mathbf{R}(q)k_q)v_q^\top$. When ϕ is a pointwise nonlinearity, the rotation and the nonlinearity generally do not commute, so the summary can depend on absolute positions in a way that cannot be cleanly absorbed into a relative offset. This mismatch motivates RoLA: in the low-rank branch, RoPE is applied after the low-rank nonlinearity so that position information is injected without destroying the reusable global summary.

3 METHOD

The proposed method decomposes attention into two complementary branches: a hardware-efficient block-sparse branch that captures high-energy semantic spikes, and a low-rank linear branch that reconstructs the smooth global background continuum. The sparse branch follows standard block-sparse attention, e.g., SLA/VMoBA (Zhang et al., 2025a; Wu et al., 2025), and is orthogonal to our main contribution; we therefore focus on the low-rank branch. In experiments, we keep this sparse branch fixed across compared sparse-based methods unless otherwise stated.

Why keep a sparse-plus-global decomposition? Several efficient-attention formulations view 3D-RoPE video attention as sharp sparse spikes plus a smoother compressible background (Liu et al., 2026b; Zhang et al., 2025a; Wu et al., 2025). We adopt this shared decomposition and therefore do not claim it as our contribution; instead, we focus on a narrower question: *what should the compressed global branch look like if it must remain linear-time while preserving compatibility with relative rotary geometry?* Our contribution is the *form* of the low-rank branch: instead of replacing cross-token aggregation with an absolute-coordinate surrogate, we keep a genuine linear-attention branch and make it compatible with relative rotary geometry by combining RoPE outside the activation (Section 3.3) with rank-truncated reuse of the pre-trained rotary schedule (Section 3.2), motivated by the formal insights in Appendices C.1 and C.3. Table 1 summarizes this design space: the sparse local path and the global correction are shared with prior designs, while our distinction lies in retaining genuine cross-token global routing, preserving relative geometry in that path, and avoiding an auxiliary positional module.

3.1 ROTARY-POSITIONED LOW-RANK LINEAR ATTENTION

Let $q_p^{\mathrm{n}} = \mathrm{norm}_q(q_p)$ and $k_q^{\mathrm{n}} = \mathrm{norm}_k(k_q)$ denote the main-path normalized, *pre-RoPE* query and key representations, where norm_q and norm_k denote the query/key normalization used in the

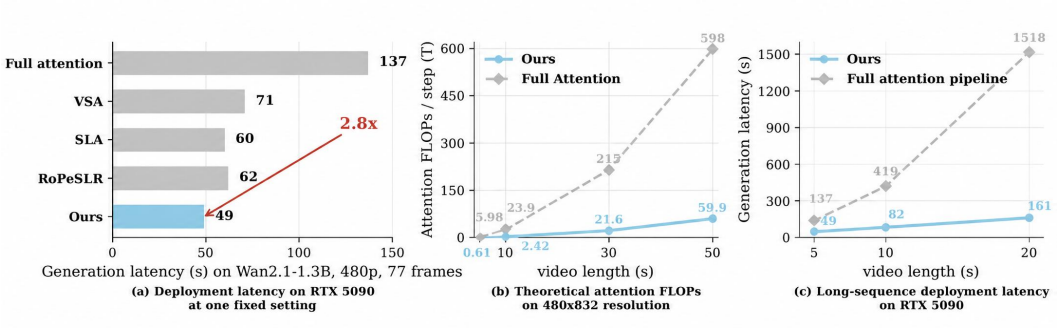


Figure 1: Deployment-oriented efficiency summary on RTX 5090 (different from the benchmark evaluation setting). (a) fixed-setting deployment latency, (b) theoretical self-attention FLOPs per denoising step, (c) long-sequence deployment latency.

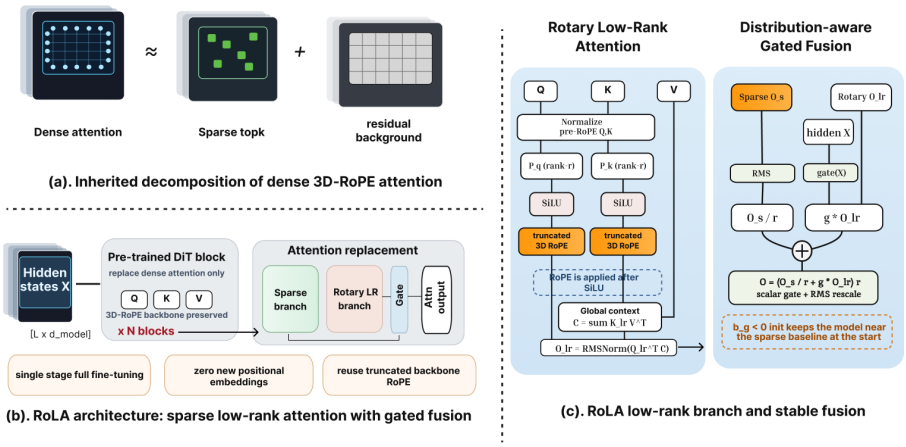


Figure 2: Architecture overview. (a) Dense 3D-RoPE attention decomposes into sparse top-k spikes and a residual background. (b) Dense attention in pre-trained DiT blocks is replaced by sparse–low-rank branches with gated fusion, reusing backbone RoPE without extra positional embeddings; right: the rotary low-rank branch and distribution-aware gated fusion.

Table 1: Capability-level comparison of representative efficient-attention methods. ✓ / × / ≈ denote present, absent, and approximate. †RoPeSLR's Fourier compensator uses absolute coordinates as a relative-position surrogate.

Method	Sparse local path	Global compensation	Genuine global routing	Relative-aware global path	Reusable linear summary	No extra positional params
Pure sparse	✓	×	×	×	×	✓
VSA	✓	×	×	×	×	✓
SLA	✓	✓	✓	×	✓	✓
RoPeSLR	✓	✓	×	≈†	×	×
RoLA (ours)	✓	✓	✓	✓	✓	✓

pre-trained backbone. The low-rank branch computes

$$\tilde{q}_p = \mathbf{R}_r(p)\ \mathrm{SiLU}\big(P_q\, q_p^{\mathrm{n}}\big), \qquad \tilde{k}_q = \mathbf{R}_r(q)\ \mathrm{SiLU}\big(P_k\, k_q^{\mathrm{n}}\big), \tag{2}$$

and

$$C = \sum_{q=1}^{L} \tilde{k}_q\, v_q^{\top} \ \in\ \mathbb{R}^{r\times d_h}, \qquad O_p^{\mathrm{lr}} = \mathrm{RMSNorm}\big(\tilde{q}_p^{\top} C\big), \tag{3}$$

where $P_q, P_k \in \mathbb{R}^{r\times d_h}$ are head-specific projections, v_q is the backbone value vector and is not rotated, and $\mathbf{R}_r(p)$ is the 3D RoPE rotation restricted to the first r coordinates (Section 3.2). The global context C is computed *once* per head, so the branch costs $\mathcal{O}(Lrd_h)$, i.e., it is linear in L. We use RMSNorm instead of the standard linear attention denominator ($\sum_q \tilde{k}_q$) because the rotated keys can have both positive and negative components, causing the sum to approach zero and introducing numerical instability. RMSNorm is parameter-free (no learnable scale or bias) and serves purely as a stabilization layer. Our relative-position analysis concerns the pre-RMSNorm logits; RMSNorm acts as a stabilization layer on the readout channels.

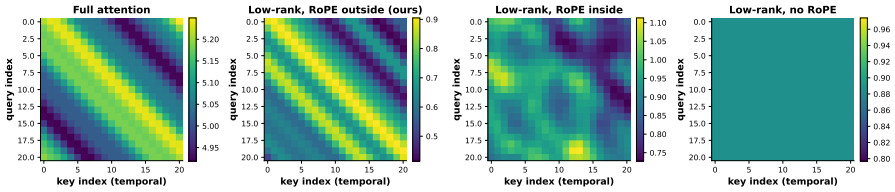


Figure 3: Content-controlled positional kernels on a temporal slice. Left to right: full attention, and the low-rank branch with RoPE applied *outside* / *inside* the activation, and without RoPE. Only the outside-activation design matches the banded relative-distance pattern of full attention; the inside variant distorts it, and removing RoPE collapses the map to a position-blind response. Color bars are per-panel, so the comparison targets spatial pattern rather than absolute ranges.

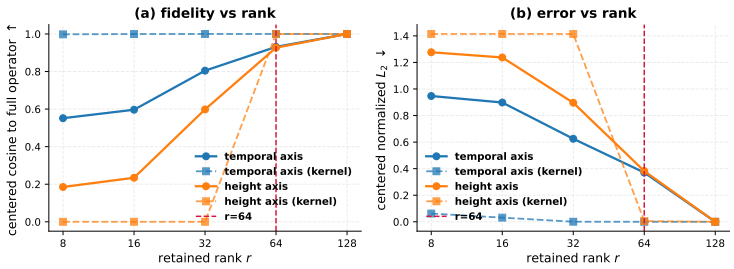


Figure 4: Fidelity of the rank-r truncated RoPE operator relative to the full operator ($r = 128$), measured by centered cosine similarity (left) and normalized L_2 error (right) as r increases. Solid lines use real-content operators, dashed lines use content-controlled kernels; blue/orange are temporal/height slices. Both metrics saturate by $r = 64$, motivating our default rank.

Two properties distinguish Eqs. (2) and (3) from both standard linear attention and the low-rank Fourier compensator used by Liu et al. (2026b). First, the rotation $\mathbf{R}_r$ is applied to the low-rank feature rather than to the full head dimension. Second, it is applied *after* the nonlinearity SiLU. The full forward pass, including the sparse branch and gated fusion, is given in Algorithm 1 (Section B). When r is close to $d_h/2$, the leading matmul cost of the low-rank branch can be comparable to that of a full-width linear branch (Section G); the purpose of the design is therefore not to reduce the linear-branch constant, but to obtain a relative-position-aware global branch that remains reusable under high sparsity.

3.2 Dimensionality Truncation of the Rotary Schedule

Why truncate the rotary schedule? The projected feature $\mathrm{SiLU}(P_q q_p^{\mathrm{n}}) \in \mathbb{R}^r$ lives in an r-dimensional space with $r \ll d_h$; we use $r = 64$ by default. Since RoPE acts on 2D frequency pairs, the low-rank branch can carry only $r/2$ rotary frequencies and therefore cannot reuse the full d_h-dimensional schedule. We truncate the pre-trained schedule to the first r coordinates under the backbone's RoPE coordinate ordering. These coordinates correspond to $r/2$ complete rotary pairs, and we reuse the backbone's own $(\mathrm{freqs_cos}, \mathrm{freqs_sin})$ tensors. This adds no positional parameters and keeps the low-rank branch on the same geometric scale as the sparse branch and the pre-trained backbone. The Wan2.1 backbone uses 3D RoPE with equal allocation across axes: for $d_h = 128$, the dimensions are partitioned as $(d_t, d_h, d_w) = (42, 43, 43)$ for temporal, height, and width respectively. Truncating to $r = 64$ retains the full temporal axis and partial height/width information. Because 3D RoPE concatenates the per-axis frequencies in a fixed order, this native truncation retains proportionally more of the leading axes and fewer of the trailing ones; the low-rank branch is intended to model the smooth global background, whose fine-grained per-axis relative structure is complemented by the sparse branch, so a reduced rotary budget on some axes is by design rather than an omission. We report representative temporal and height slices in the mechanism experiments and validate the retained operator empirically (Fig. 4).

Why is $r = 64$ sufficient in practice? The rank-r truncation is motivated by a simple error intuition: if the retained rotary coordinates capture most of the positional structure relevant to the global branch, the discarded rotary coordinates contribute only a small residual score. Section C.2 gives an idealized nested-setting intuition for why such truncation can be benign. Because the trained $\mathbf{P}_q, \mathbf{P}_k$ are learned directly in the low-rank space, this intuition should not be read as a guarantee for the trained branch; the empirical fidelity sweep below is the operative evidence. This intuition is consistent with the sparse-plus-smooth background view associated with low-rank Fourier compensation (Liu et al., 2026b): the global branch mainly models a smooth and compressible background, suggesting that its positional structure can be captured in a reduced rotary subspace. In our implementation, we retain the first r rotary coordinates, i.e., $r/2$ complete rotary coordinate pairs, in the backbone's native ordering; we do not reorder these coordinates by frequency. Empirically, Fig. 4 evaluates this design choice: the centered cosine similarity rises and the normalized L_2 error falls monotonically, with both metrics saturating by $r = 64$. Concretely, the temporal cosine improves from 0.55 to 0.80 and 0.93 at r=$8, 32, 64$, while the height cosine improves from 0.19 to 0.60 and 0.93. We therefore choose $r = 64$ as the default operating point.

3.3 RoPE Outside the Activation

A key design choice in the low-rank branch concerns the *order* of the rotation and the nonlinearity. In RoLA, we first apply the low-rank projection and the pointwise nonlinearity, and then apply the truncated RoPE rotation. This ordering is motivated by a simple separation of roles: the nonlinear low-rank map should extract position-free content features, while RoPE should act only as a relative-position operator. When rotation is applied after the nonlinearity, the global context can still be accumulated once and reused by all queries; the positional effect appears only when a query reads out this context. Appendix C.1 gives a formal statement of this relative-position structure.

If RoPE is instead applied before the nonlinearity, the rotation and the pointwise activation no longer commute. As a result, position information becomes entangled with the content features before the global aggregation, and the resulting key-side summary can depend on absolute positions rather than only on relative offsets. Appendix C.3 formalizes this obstruction and explains why the inside-activation ordering cannot generally admit an exact relative linear-attention factorization. Our mechanism experiments in Section 4.1 test these two predictions directly.

Remark 1 (Contrast with absolute-PE injection)**.** *The low-rank Fourier compensator of Liu et al. (2026b) sidesteps the dilemma by removing cross-token routing altogether and re-injecting position as a learnable absolute embedding. It learns to imitate relative decay from absolute coordinates. RoLA instead keeps genuine cross-token routing and obtains relative positional behavior through the outside-activation rotary design. The two designs are complementary points in the design space; we compare them empirically in Section 4.3.*

3.4 Feature Alignment and Gated Fusion

Fusing localized sparse outputs O^{s} with the global low-rank output O^{lr} requires variance stabilization. We normalize with RMSNorm (Zhang & Sennrich, 2019) and combine the two branches through a token-wise scalar gate $g = \sigma(XW_g + b_g) \in (0,1)^{L\times 1}$, where X denotes the input hidden states of the attention block and $W_g \in \mathbb{R}^{d_{\mathrm{model}}\times 1}$. The gate is deliberately bottlenecked to one scalar per token, so it modulates the magnitude of the low-rank branch rather than changing its channel-wise direction. The RMS of the sparse output provides a per-token scale reference for stable fusion:

$$r_p = \sqrt{\tfrac{1}{d_h}\textstyle\sum_i (O^{\mathrm{s}}_{p,i})^2 + \epsilon}, \qquad O_p = \left(\frac{O^{\mathrm{s}}_p}{r_p} + g_p\, O^{\mathrm{lr}}_p\right) r_p. \tag{4}$$

The sparse-output RMS is a natural per-token scale reflecting the magnitude of the branch active at that token: dividing by r_p normalizes the sparse branch, and multiplying back restores the token scale after adding the gated global correction. The gate bias b_g is initialized to a negative value so that the branch starts close to the sparse baseline, allowing the low-rank contribution to be introduced gradually during alignment.

3.5 Implementation Details

Post-training strategy. RoLA is integrated into a pre-trained DiT by a single-stage full fine-tuning procedure. We initialize the new parameters $\Theta_{\mathrm{new}} = \{P_q, P_k, W_g, b_g\}$ with Kaiming initialization (He et al., 2015) for the projections, zero for W_g, and a negative b_g so that the module starts close to the sparse baseline. We then replace the attention module with RoLA and jointly optimize all parameters, including the backbone and Θ_{new}, under the standard diffusion flow-matching objective $\mathcal{L}_{\mathrm{task}}$ (Lipman et al., 2022). The adaptation is lightweight relative to pre-training: only the new branch parameters are trained from scratch, while the backbone is updated with a small learning rate. The rationale for adopting this single-stage recipe, instead of the two-stage strategy associated with the low-rank Fourier compensator (Liu et al., 2026b), is discussed in Section F.1.

Engineering: fused inference kernel. At inference, the low-rank branch is evaluated using two fused Triton kernels. Because RoPE acts on even/odd coordinate pairs, we split each projection weight into even and odd rows offline and process the two halves as pure 2D matmuls, keeping the computation as dense matrix multiplications and avoiding irregular indexing. Kernel 1 accumulates the even/odd context matrices $C_{\mathrm{even}} = \sum_q \tilde{k}^{\mathrm{even}}_q v_q^\top$ and $C_{\mathrm{odd}} = \sum_q \tilde{k}^{\mathrm{odd}}_q v_q^\top$ over key/value tiles, and Kernel 2 forms $O^{\mathrm{lr}}_p = \tilde{q}^{\mathrm{even}}_p C_{\mathrm{even}} + \tilde{q}^{\mathrm{odd}}_p C_{\mathrm{odd}}$ and applies the per-token RMSNorm. This

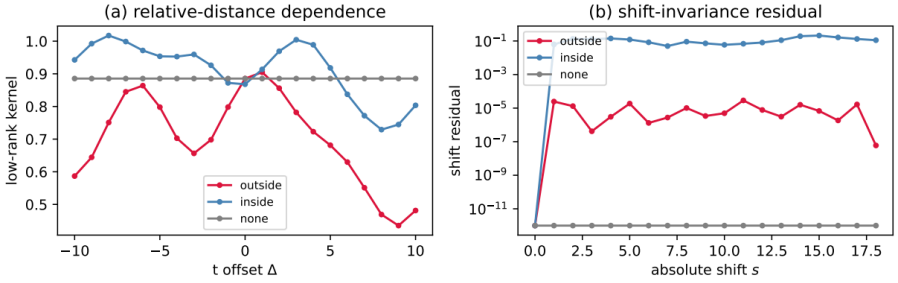


Figure 5: Effect of RoPE placement in the low-rank branch. (a) low-rank kernel vs. relative offset (tests position-awareness); (b) shift-invariance residual under absolute translation (tests whether the dependence is purely relative). Only RoPE *outside* the activation satisfies both: non-trivial offset dependence in (a) and near-zero shift residual in (b).

implementation exactly reproduces Eqs. (2) and (3). Detailed FLOPs derivations and asymptotic complexity analysis are provided in Section G.

4 Experiments

We evaluate RoLA at both the mechanism level and the generation level. Mechanism experiments test the two design principles of the method, while generation experiments measure quality and efficiency on full-scale video models.

4.1 Mechanism-Level Validation

Protocol. Mechanism experiments use pre-trained Wan2.1-T2V-1.3B (Wan et al., 2025) weights with $d_h = 128$ and $r = 64$. We extract normalized query/key features from real 480p, 81-frame denoising trajectories, average over heads and denoising steps, and use the low-rank projections from the single-stage fine-tuned checkpoint. For all positional metrics, we use pre-RMSNorm low-rank logits.

Low-rank branch encodes useful positional structure only with outside-RoPE (Fig. 3). We construct the low-rank attention map for a contiguous temporal slice and compare it with the full-attention positional kernel under content-controlled inputs. The full kernel exhibits a clear banded distance-decay structure. When RoPE is applied *outside* the activation, the low-rank branch closely matches this banded relative-position pattern. In contrast, the inside-activation variant remains structured but visibly distorts the clean relative geometry of the full kernel. Removing RoPE collapses the map to a nearly flat, position-blind response. Thus RoPE is necessary for the low-rank branch to encode positional structure, and only the outside placement preserves the desired relative-position geometry.

Table 2: Positional fidelity by RoPE ordering.

RoPE ordering	Shift residual ↓	Offset range (position-aware)	Behavior
No RoPE	0.0	0.0	position-blind
Inside activation	3.7×10^{-1}	0.47	leaks absolute position
Outside activation (ours)	1.8×10^{-5}	0.47	**relative / shift-invariant**

RoPE placement: outside is both position-aware and shift-invariant (Fig. 5 and Table 2). For each of the three orderings—rotation *outside* the activation (ours), rotation *inside* the activation, and *no* rotation—we measure: (i) the relative-distance dependence of the low-rank logit as a function of the temporal offset $\Delta = p - q$; and (ii) the shift-invariance residual $|s(p+\delta, q+\delta) - s(p, q)|$ at a fixed offset as the pair is translated. An ordering that preserves relative-position structure should exhibit both a non-trivial dependence on Δ and a near-zero shift residual. Only the outside-activation ordering satisfies both conditions: it achieves a shift residual of 1.8×10^{-5}, which indicates near-zero translation dependence, while maintaining an offset range of 0.47 (Table 2). Rotating *inside* the activation produces similar offset variation but a much larger shift residual (3.7×10^{-1}), consistent with the absolute-position leakage analyzed in Section C.3. The no-RoPE branch has zero shift residual only trivially, because it is entirely position-blind with offset range 0.0; we therefore require both non-zero offset dependence and near-zero shift residual.

Table 4: Quantitative comparison on Wan2.1-1.3B (480p) and Wan2.1-14B (720p, 81 frames), all on NVIDIA H100 80G GPUs. Latency is measured for the DiT denoising component under the VBench setting, and is therefore not directly comparable to the RTX 5090 deployment numbers in Figure 1.

Model	Method	VBench Quality					Additional Metrics			Efficiency	
		SC↑	MS↑	BC↑	IQ↑	AQ↑	PSNR↑	SSIM↑	LPIPS↓	Latency↓	Sparsity↑
Wan2.1-1.3B	Full	0.9403	0.9557	**0.9498**	**0.6510**	**0.6088**	–	–	–	50.9s	0%
	VSA	0.9364	0.9757	0.9421	0.6328	0.5629	23.05	0.819	0.149	34.3s	90%
	SALAD	0.9427	0.9768	0.9427	0.6312	0.5932	24.19	0.825	0.136	48.5s	90%
	SLA	0.9401	0.9782	0.9486	0.6343	0.5918	24.17	0.844	0.130	31.2s	90%
	RoPeSLR	0.9512	0.9789	0.9485	0.6481	0.6010	25.01	0.843	0.129	30.4s	90%
	RoLA (Ours)	**0.9521**	**0.9812**	0.9490	0.6492	0.6068	**25.67**	**0.860**	**0.121**	**21.7s**	90%
Wan2.1-14B	Full	**0.9698**	0.9850	**0.9681**	**0.6996**	**0.6319**	–	–	–	1075.7s	0%
	VSA	0.9010	0.9788	0.9482	0.6012	0.5998	24.27	0.843	0.130	717.1s	90%
	SALAD	0.9487	0.9796	0.9476	0.6371	0.6054	26.11	0.873	0.126	1024.5s	90%
	SLA	0.9481	0.9802	0.9511	0.6398	0.6052	26.02	0.878	0.124	601.0s	90%
	RoPeSLR	0.9601	0.9842	0.9528	0.6627	0.6102	26.33	0.880	0.118	598.3s	90%
	RoLA (Ours)	0.9629	**0.9853**	0.9537	0.6783	0.6294	**28.65**	**0.891**	**0.108**	**409.3s**	90%

Table 3: Rank-truncation fidelity.

Retained rank r	**8**	**16**	**32**	**64 (default)**	**128**
Cosine to full, temporal ↑	0.55	0.60	0.80	0.93	1.00
Cosine to full, height ↑	0.19	0.24	0.60	0.93	1.00

Rank truncation approximates the full operator (Fig. 4 and Table 3). To test whether truncation preserves the relative-position operator, we sweep $r \in \{8, 16, 32, 64, 128\}$ and compare each rank-r operator with the full ($r = 128$) reference on real Wan Q/K features. For each axis-specific slice A_r and reference A_{128}, we mean-center them to obtain $\widetilde{A}_r$ and $\widetilde{A}_{128}$, and measure their centered cosine similarity and normalized L_2 error. Both metrics improve and saturate near $r = 64$: the temporal cosine rises from 0.55 to 0.80 and 0.93 at r=$8, 32, 64$, while the height cosine rises from 0.19 to 0.60 and 0.93; normalized L_2 error falls near zero. This supports $r = 64$ as a sufficient rank, consistent with Section C.2.

4.2 Generation Quality

Models and datasets. We evaluate RoLA on Wan2.1-1.3B and Wan2.1-14B (Wan et al., 2025). For lightweight adaptation, we use curated subsets of OpenSora (Zheng et al., 2024) and OpenVid-1M (Nan et al., 2024), respectively. Quantitative evaluation follows the official VBench protocol; complete results are provided in Appendix Table 8.

Baselines. We compare with dense full attention, VSA (Zhang et al., 2025b), SLA (Zhang et al., 2025a), SALAD (Fang et al., 2026), and RoPeSLR (Liu et al., 2026b). The dense baseline uses the official Wan2.1 implementation with FlashAttention-2 (Dao, 2024) enabled. All methods use their official implementations and available kernel optimizations; sparse baselines are evaluated at matched sparsity, while trainable baselines use the same adaptation data, batch size, and optimization steps as RoLA, retaining their method-specific routing and training recipes. SANA-Video (Chen et al., 2026b) and SANA-Video 2.0 (Chen et al., 2026a) follow a linear-video-DiT training setting rather than sparse post-training and are therefore excluded.

Metrics. We report five representative VBench dimensions (Huang et al., 2024): Subject Consistency, Motion Smoothness, Background Consistency, Imaging Quality, and Aesthetic Quality. These dimensions capture temporal stability, background consistency, and visual quality, which are most sensitive to aggressive attention sparsification. VBench measures absolute generation quality, while PSNR, SSIM, and LPIPS are computed against dense-attention outputs under matched prompts and seeds to measure relative fidelity to the dense model. Efficiency is measured by DiT denoising latency, attention sparsity, and theoretical attention cost; latency excludes text encoding and VAE stages, and sparsity denotes the fraction of query-key interactions removed.

Quality evaluation. As shown in Table 4, RoLA reduces latency over dense full attention with only a small degradation in the reported VBench dimensions. At matched 90% sparsity, it better preserves these dimensions than VSA, SALAD, SLA, and RoPeSLR. Compared with RoPeSLR (Liu et al., 2026b), which approximates relative decay using an absolute positional module, RoLA directly injects relative-position structure without extra positional parameters and remains closer to dense attention. These results agree with the mechanism-level analysis in Section 4.1.

Fine-tuning control. To isolate the effect of the attention replacement from fine-tuning itself, we additionally fine-tune the full-attention model using the same protocol as RoLA. Appendix Table 9 reports this controlled comparison, where the only difference is whether the attention block remains dense or is replaced by RoLA.

4.3 Ablation Study

We ablate the central design axes to isolate their contributions. All ablations use Wan2.1-1.3B at matched 90% sparsity.

Controlled global-branch comparison. To separate the effect of the global branch from the sparse attention pattern, we fix the sparse engine, pattern, and target sparsity, and replace only the global branch. The resulting comparison is reported in Appendix Table 10. This experiment directly tests whether the gains come from the rotary low-rank construction rather than from a different sparse routing pattern.

Table 5: Ablation on retained rotary rank r.

Rank r	SC↑	MS↑	BC↑	IQ↑	AQ↑
8	0.8788	0.8715	0.9170	0.5574	0.5640
16	0.8910	0.8861	0.9298	0.5772	0.5601
32	0.9398	0.9540	0.9394	0.6138	0.5990
64 (default)	0.9521	0.9812	0.9490	**0.6492**	0.6068
128	**0.9540**	**0.9841**	**0.9525**	0.6479	**0.6099**

Sparsity-quality trade-off. We also evaluate RoLA across multiple sparsity levels and report the selected VBench dimensions in Appendix Table 11. This sweep tests whether the proposed global branch remains useful as the sparse branch becomes increasingly aggressive.

RoPE placement (motivated by Section C.1). We compare no rotation, rotation *inside* the activation, and rotation *outside* the activation. Table 2 shows that only the outside ordering preserves relative-position structure; the inside ordering leaks absolute position despite remaining position-aware.

Rotary truncation rank (Fig. 4). We sweep $r \in \{8, 16, 32, 64, 128\}$. Operator fidelity saturates at $r = 64$ (Table 3), and Table 5 shows that downstream quality follows the same trend, with clear gains up to $r = 64$ and only marginal change beyond it.

Table 6: Activation ablation. Projections are retrained per activation; the sparse-only baseline rel-L_2 is 0.612.

Activation	Align rel-L_2 ↓
Identity (linear)	0.525
Tanh	0.525
ReLU	0.534
GELU	0.522
SiLU (default)	**0.521**

Activation function. We use *Align rel-L_2* to measure how well different activation choices allow the sparse–low-rank branch to reconstruct the dense full-attention output. For each activation, we retrain the low-rank projections and compute the relative L_2 distance $\|O^{\text{fused}} - O^{\text{full}}\|_2 / \|O^{\text{full}}\|_2$ on real Wan Q/K/V features, averaged over tokens and heads (lower is better). Table 6 shows that SiLU achieves the lowest *Align rel-L_2*. Identity underperforms, supporting the need for a nonlinear transformation; Tanh also underperforms despite being smooth, suggesting that the specific shape of the nonlinearity affects the expressiveness of the low-rank approximation.

5 Conclusion

We present RoLA, a rotary-positioned low-rank attention branch for efficient video Diffusion Transformers. By applying rank-truncated 3D RoPE outside the nonlinear activation, it resolves the compatibility between reusable linear aggregation and 3D relative rotary geometry in the low-rank global branch, keeping a reusable linear-time global summary while injecting relative positional structure without auxiliary absolute-position parameters. Mechanistic analysis and empirical evaluations indicate that RoLA achieves a favorable quality–efficiency trade-off under high sparsity, retaining generation fidelity while substantially reducing computational overhead.

Limitations. Our evaluation focuses on two Wan models under the standard sequence lengths used for this class of video generators, and the theoretical analysis provides a structural verification (Proposition 1), an idealized truncation bound relying on operator-norm fidelity under controlled inputs (Proposition 2), and an exact-factorization obstruction (Proposition 3); none formally connects to generation metrics. Code will be released.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools (Claude) for literature review assistance, drafting intermediate text, and code debugging. We did not use generative AI to generate core experimental results, mathematical derivations, or novel algorithmic ideas. All AI-assisted text was manually reviewed, verified, and revised by the authors. We take full responsibility for the final content of this work, including all claims, proofs, and experimental results.

ETHICS STATEMENT

(This section is **recommended** and does not count toward the page limit.)

This work presents a computational efficiency improvement for video diffusion models. It does not involve human subjects, sensitive data, or introduce novel dual-use concerns beyond those inherent to generative video models in general.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

To ensure reproducibility, we provide the following: (1) complete hyperparameter settings in Appendix Table 14; (2) detailed algorithm pseudocode in Appendix Algorithm 1; (3) formal analysis and proofs in Appendices C.1, C.2, and C.3; (4) evaluation is conducted on publicly available Wan2.1 models; (5) code will be released upon acceptance. All experiments were conducted on NVIDIA H100 and RTX 5090 GPUs with batch sizes and training steps detailed in the appendix. We use fixed random seeds (seed=42 for training, seed=1234 for inference) for deterministic results.

REFERENCES

Junsong Chen, Jincheng Yu, Yitong Li, Shuchen Xue, Haozhe Liu, Jingyu Xin, Yuyang Zhao, Tian Ye, Zhangjie Wu, Zian Wang, et al. Sana-video 2.0: Hybrid linear attention with attention residuals for efficient video generation. *arXiv preprint arXiv:2607.21553*, 2026a.

Junsong Chen, Yuyang Zhao, Jincheng Yu, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Zhou Daquan, Huan Ling, et al. Sana-video: Efficient video generation with block linear diffusion transformer. In *International Conference on Learning Representations*, volume 2026, pp. 38973–38997, 2026b.

Pengtao Chen, Xianfang Zeng, Maosen Zhao, Mingzhu Shen, Wei Cheng, Gang Yu, and Tao Chen. Sparse-vdit: Unleashing the power of sparse attention to accelerate video diffusion transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2957–2965, 2026c.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, volume 2024, pp. 35549–35562, 2024.

Hangliang Ding, Dacheng Li, Runlong Su, Peiyuan Zhang, Zhijie Deng, Ion Stoica, and Hao Zhang. Efficient-vdit: Efficient video diffusion transformers with attention tile. *arXiv preprint arXiv:2502.06155*, 2025.

Tongcheng Fang, Hanling Zhang, Ruiqi Xie, Zhuo Han, Xin Tao, Tianchen Zhao, Pengfei Wan, Wenbo Ding, Wanli Ouyang, Xuefei Ning, et al. Salad: Achieve high-sparsity attention via efficient linear attention tuning for video diffusion transformer. *arXiv preprint arXiv:2601.16515*, 2026.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

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

Weijie Kong, Qi Tian, Zijian Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint*, 2024.

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

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

Yuxi Liu, Zekun Zhang, Yixiang Cai, Renjia Deng, Yutong He, and Kun Yuan. RoPeSLR: 3d RoPE-driven sparse-lowrank attention for efficient diffusion transformers, 2026b. URL `https://arxiv.org/abs/2605.20659`.

Kepan Nan, Rui Xie, Penghao Zhou, et al. OpenVid-1M: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023.

Zhen Qin, Weixuan Sun, Kaiyue Lu, Hui Deng, Dongxu Li, Xiaodong Han, Yuchao Dai, Lingpeng Kong, and Yiran Zhong. Linearized relative positional encoding. *arXiv preprint arXiv:2307.09270*, 2023.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Team Wan et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint*, 2025.

Jianzong Wu, Liang Hou, Haotian Yang, et al. VMoBA: Mixture-of-block attention for video diffusion models. *arXiv preprint*, 2025.

Haocheng Xi, Shuo Yang, Yilong Zhao, et al. Sparse VideoGen: Accelerating video diffusion transformers with spatial-temporal sparsity. In *International Conference on Machine Learning (ICML)*, 2025.

Shuo Yang, Haocheng Xi, Yilong Zhao, et al. Sparse VideoGen2: Accelerate video generation with sparse attention via semantic-aware permutation. *arXiv preprint*, 2025a.

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.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Jintao Zhang, Haoxu Wang, Kai Jiang, et al. SLA: Beyond sparsity in diffusion transformers via fine-tunable sparse-linear attention. *arXiv preprint*, 2025a.

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

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, et al. VSA: Faster video diffusion with trainable sparse attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, et al. Open-sora: Democratizing efficient video production for all. *arXiv preprint*, 2024.

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

A Related Work

Sparse attention for video DiTs. Sparse attention is a dominant approach to the quadratic attention bottleneck in video DiTs, exploiting structured, learned, or dynamic sparsity in spatiotemporal interactions. Sparse VideoGen (Xi et al., 2025) and SVG2 (Yang et al., 2025a) exploit spatial-temporal sparsity with semantic-aware token reorganization; Efficient-vDiT (Ding et al., 2025) leverages repetitive attention tiles, while Sparse-vDiT (Chen et al., 2026c) identifies recurring structured sparsity patterns across heads and layers. VSA (Zhang et al., 2025b) learns trainable sparse patterns, VMoBA (Wu et al., 2025) organizes attention into a mixture of blocks, Radial Attention (Li et al., 2026) exploits spatiotemporal energy decay for distance-aware sparse computation, and MOD-DiT (Liu et al., 2026a) dynamically predicts sparsity patterns across denoising intervals. Despite substantial efficiency gains, these sparse-only approaches retain only a subset of query-key interactions; under aggressive sparsity, weak but collectively important long-range context can be discarded, motivating complementary global modeling.

Sparse linear hybrids. To recover global context discarded by sparsification, recent methods combine sparse attention with efficient linear branches. SLA (Zhang et al., 2025a), SLA2 (Zhang et al., 2026), SALAD (Fang et al., 2026), and SVG-EAR (Zhou et al., 2026) explore trainable or lightweight sparse–linear compensation for discarded global interactions. Conventional sparse–linear formulations can nevertheless encounter the RoPE Dilemma (Section 2): nonlinear feature maps generally do not commute with rotary transformations, making it difficult to preserve 3D-RoPE relative-position structure in a reusable linear branch. The limitation becomes more critical at extreme sparsity, where the linear branch must recover more discarded interactions. Related linear video DiTs, including SANA-Video (Chen et al., 2026b) and SANA-Video 2.0 (Chen et al., 2026a), further introduce RoPE-aware and hybrid linear-attention designs. We treat these SANA-style models as a separate pure linear-pretraining line rather than as direct baselines, because our main comparison targets sparse post-training hybrids under the same video-DiT adaptation setting. RoPeSLR, a low-rank Fourier compensator (Liu et al., 2026b), instead sidesteps the dilemma in sparse–low-rank attention by replacing cross-token linear aggregation with a per-token coordinate MLP and a learnable absolute 3D positional embedding. Building on this sparse–global decomposition, our method retains genuine cross-token linear aggregation while directly reusing the pre-trained 3D-RoPE structure in the low-rank branch.

Rotary position embeddings. RoPE (Su et al., 2021) encodes relative position through orthogonal rotations with an exponentially decaying frequency schedule. The compatibility of relative positional encoding with linear attention has also been studied in LRPE (Qin et al., 2023), which preserves relative positional modeling under linear-complexity computation. In video DiTs, however, the head dimension is split across the three spatiotemporal axes (3D RoPE), so the per-axis frequency budget is small; we reuse the first r rotary coordinates in the backbone's native ordering, i.e., $r/2$ complete rotary coordinate pairs supported by the low-rank bottleneck, provide an idealized error-analysis intuition for this reuse (Section C.2), and validate the resulting operator empirically (Section 3.2; Fig. 4). Unlike LRPE, RoLA is not a generic standalone positional encoder; it is a sparse-video-DiT construction that preserves a reusable cross-token global branch while remaining compatible with a pre-trained 3D-RoPE backbone and the sparse engine.

B Full Forward Pass

Algorithm 1 gives the complete RoLA attention forward pass for a single block, combining the sparse branch, the rotary-positioned low-rank branch, and the distribution-aware gated fusion of Section 3. All new parameters are jointly optimized with the backbone.

C Theoretical Insights and Proof Sketches

This appendix collects formal insights that motivate the design of RoLA. These results are not intended as full guarantees for end-to-end generation quality; rather, they explain the structural properties of the low-rank branch and justify the two design choices used in the main text: applying RoPE outside the nonlinear feature map, and truncating the rotary schedule to the low-rank bottleneck.

Algorithm 1 RoLA attention forward pass (single block). Block-level operations (Lines 1, 8, 11) are performed once; per-head operations (Lines 2–7, 9–10) are repeated for each head h.

Require: hidden states $X \in \mathbb{R}^{L\times d_{\text{model}}}$; attention module with projections W_q, W_k, W_v, pre-attention norms $\text{norm}_q, \text{norm}_k$, output projection W_o; new parameters $P_q, P_k \in \mathbb{R}^{r\times d_h}$, gate $W_g \in \mathbb{R}^{d_{\text{model}}\times 1}$, gate bias b_g; full rotary schedule $\mathbf{R}(\cdot)$; rank-truncated schedule $\mathbf{R}_r(\cdot)$; sparse engine SparseAttn (e.g., VMoBA/SLA) with topk ratio

Ensure: attention output $O \in \mathbb{R}^{L\times d_{\text{model}}}$

1: $Q, K, V \leftarrow W_q X,\ W_k X,\ W_v X$ ▷ QKV projection

2: $Q^{\text{n}}, K^{\text{n}} \leftarrow \text{norm}_q(Q),\ \text{norm}_k(K)$ ▷ pre-RoPE normalized query/key, reshaped to $[L, d_h]$ per head

– Sparse branch (high-energy spikes) –

3: $Q^{\text{r}}, K^{\text{r}} \leftarrow \mathbf{R}(Q^{\text{n}}),\ \mathbf{R}(K^{\text{n}})$ ▷ full-dimensional RoPE on the main path

4: $O^{\text{s}} \leftarrow \text{SparseAttn}(Q^{\text{r}}, K^{\text{r}}, V)$ ▷ $\mathcal{O}\big(L^2(1-S)d_h\big)$

– Low-rank branch (smooth global background), RoPE outside activation –

5: $\tilde{q} \leftarrow \mathbf{R}_r\big(\text{SiLU}(Q^{\text{n}}P_q^\top)\big),\quad \tilde{k} \leftarrow \mathbf{R}_r\big(\text{SiLU}(K^{\text{n}}P_k^\top)\big)$ ▷ activate, then rotate; $\tilde{q}, \tilde{k} \in \mathbb{R}^{L\times r}$

6: $C \leftarrow \tilde{k}^\top V \in \mathbb{R}^{r\times d_h}$ ▷ global context, computed once per head: $\mathcal{O}(Lrd_h)$

7: $O^{\text{lr}} \leftarrow \text{RMSNorm}\big(\tilde{q}\, C\big)$ ▷ per-query readout: $\mathcal{O}(Lrd_h)$

– Distribution-aware gated fusion –

8: $g \leftarrow \sigma\big(XW_g + b_g\big) \in (0,1)^{L\times 1}$ ▷ token-wise scalar gate

9: $\rho \leftarrow \sqrt{\frac{1}{d_h}\sum_i (O^{\text{s}}_{:,i})^2 + \epsilon}$ ▷ per-token RMS of the sparse output

10: $O^{\text{fused}} \leftarrow \big(O^{\text{s}}/\rho + g\, O^{\text{lr}}\big)\,\rho$ ▷ RMS-normalized gated combination, Eq. 4

11: $O \leftarrow W_o\, \text{concat}_h\big(O^{\text{fused}}\big)$ ▷ merge heads, output projection

12: **return** O

Throughout this appendix, $\mathbf{R}_r(p) \in \mathbb{R}^{r\times r}$ denotes the 3D RoPE rotation restricted to the first r rotary coordinates in the backbone's native coordinate ordering, i.e., the block-diagonal matrix formed by the retained complete 2D rotary pairs, split across the three spatiotemporal axes. This notation only specifies the retained coordinate indices and does not make any frequency-ordering assumption. It is orthogonal,

$$\mathbf{R}_r(p)^\top \mathbf{R}_r(p) = I,$$

and satisfies the group property

$$\mathbf{R}_r(p)^\top \mathbf{R}_r(q) = \mathbf{R}_r(q-p),$$

which is inherited blockwise from the 2D rotation identity

$$\text{Rot}(\theta p)^\top \text{Rot}(\theta q) = \text{Rot}(\theta(q-p)).$$

C.1 Relative-Position Structure of the Outside-Activation Branch

The following insight formalizes the main reason for applying RoPE after the low-rank nonlinearity: if the nonlinear feature map is position-free, the rotary rotation can be moved into a relative offset without destroying the reusable linear aggregation.

Proposition 1 (Relative-position structure of the outside-activation branch)**.** *Let $\phi(\cdot) = \text{SiLU}(P_q\,\cdot)$ and $\psi(\cdot) = \text{SiLU}(P_k\,\cdot)$ be the pointwise low-rank feature maps. With RoPE applied* outside *the activation as in Eq.* (2)*, the low-rank pairwise interaction is a function of the relative offset:*

$$\tilde{q}_p^\top \tilde{k}_q = \big(\mathbf{R}_r(p)\phi(q_p)\big)^\top \big(\mathbf{R}_r(q)\psi(k_q)\big) = \phi(q_p)^\top\, \mathbf{R}_r(q-p)\, \psi(k_q). \tag{5}$$

Meanwhile, the global context $C = \sum_q \tilde{k}_q v_q^\top$ remains a single reusable sum. Hence the branch is simultaneously shift-invariant and $\mathcal{O}(L)$.

Proof. Write the low-rank features in Eq. (2) as

$$\tilde{q}_p = \mathbf{R}_r(p)\phi(q_p), \qquad \tilde{k}_q = \mathbf{R}_r(q)\psi(k_q),$$

where

$$\phi(q_p) = \text{SiLU}(P_q q_p^{\text{n}}), \qquad \psi(k_q) = \text{SiLU}(P_k k_q^{\text{n}})$$

are position-free: they depend only on token content, not on the indices p or q.

Step 1: relative interaction. For any query–key pair,

$$\tilde{q}_p^\top \tilde{k}_q = \big(\mathbf{R}_r(p)\phi(q_p)\big)^\top \big(\mathbf{R}_r(q)\psi(k_q)\big) = \phi(q_p)^\top \mathbf{R}_r(p)^\top \mathbf{R}_r(q)\,\psi(k_q) = \phi(q_p)^\top \mathbf{R}_r(q-p)\,\psi(k_q),$$

where the last equality uses orthogonality and the group property. The right-hand side depends on the positions only through the relative offset $q-p$, establishing shift-invariance of the low-rank logit.

Step 2: $\mathcal{O}(L)$ associativity is preserved. The global context is

$$C = \sum_{q=1}^{L} \tilde{k}_q v_q^\top = \sum_{q=1}^{L} \mathbf{R}_r(q)\psi(k_q) v_q^\top \in \mathbb{R}^{r\times d_h}.$$

This is a single sum that does not depend on the query index p; it can therefore be computed once at cost $\mathcal{O}(Lrd_h)$. The per-query output

$$O_p^{\mathrm{lr}} = \mathrm{RMSNorm}(\tilde{q}_p^\top C)$$

costs $\mathcal{O}(rd_h)$ per token, so the branch is $\mathcal{O}(Lrd_h)$ overall. The rotation $\mathbf{R}_r(q)$ is absorbed into C before the sum, and the query-side rotation $\mathbf{R}_r(p)$ is applied after C; neither obstructs associativity, because both are linear operators acting outside the position-free feature maps. This is what simultaneously yields relative-position structure and linear cost. □

C.2 Truncation-Error Intuition for the Rotary Low-Rank Branch

Notation and idealized nested view. Let $\mathbf{R}(p) \in \mathbb{R}^{d_h\times d_h}$ be the full 3D RoPE matrix, block-diagonal with 2×2 rotation blocks. Let $r = 2s \le d_h$ be an even truncation rank, and assume that the first r coordinates in the backbone's native ordering correspond to the first s complete 2×2 blocks. Define

$$P_{\le r} = \mathrm{diag}(I_r,\ 0_{(d_h-r)\times(d_h-r)}), \qquad P_{>r} = I_{d_h} - P_{\le r}.$$

For a vector $u \in \mathbb{R}^{d_h}$, write

$$u_{\le r} = P_{\le r}u, \qquad u_{>r} = P_{>r}u.$$

The truncated RoPE matrix is

$$\mathbf{R}_r(p) = P_{\le r}\mathbf{R}(p)P_{\le r}.$$

To make the truncation intuition precise, we consider an idealized nested setting. Let $\Phi(q_p)$ and $\Psi(k_q)$ be position-free full-dimensional feature maps, and suppose the rank-r branch retains the first r rotary coordinates of these maps in the backbone's native ordering. In the actual implementation, P_q and P_k are learned directly in the low-dimensional space. Therefore, the following statement is not a theoretical contribution and should not be read as a strict guarantee for the trained branch; it is an idealized design intuition. The empirical operator-fidelity results in the main text are the operative evidence.

Assumption 1 (Concentration in the retained rotary subspace)**.** *For the global background branch of a pre-trained video DiT, the residual feature components outside the retained rotary subspace are small. Specifically, there exist constants $\epsilon_\Phi(r), \epsilon_\Psi(r) > 0$ such that*

$$\sup_p \|P_{>r}\Phi(q_p)\|_2 \le \epsilon_\Phi(r), \qquad \sup_q \|P_{>r}\Psi(k_q)\|_2 \le \epsilon_\Psi(r).$$

For 3D-RoPE video DiTs, this concentration is consistent with the sparse-plus-smooth background decomposition identified in the low-rank Fourier compensation analysis of RoPESLR (Liu et al., 2026b). RoPE uses a geometrically spaced rotary frequency schedule (Su et al., 2021); our implementation retains the first r rotary coordinates in the backbone's native ordering, and this choice is justified by the empirical fidelity sweep rather than by a frequency-ordering assumption. Fig. 4 validates the sufficiency of this retained subspace at $r = 64$.

Proposition 2 (Truncation-error intuition for a nested rotary feature map)**.** *Define the full and rank-r truncated pairwise scores as*

$$s_{\mathrm{full}}(p,q) = \Phi(q_p)^\top \mathbf{R}(q-p)\Psi(k_q),$$

and

$$s_r(p,q) = \Phi_{\le r}(q_p)^\top \mathbf{R}_r(q-p)\Psi_{\le r}(k_q).$$

Then for all tokens p, q,

$$|s_{\text{full}}(p,q) - s_r(p,q)| \le \|P_{>r}\Phi(q_p)\|_2 \, \|P_{>r}\Psi(k_q)\|_2 .$$

In particular, under Assumption 1,

$$|s_{\text{full}}(p,q) - s_r(p,q)| \le \epsilon_\Phi(r)\, \epsilon_\Psi(r).$$

Remark. This bound provides a structural justification for RoPE truncation in terms of operator-norm fidelity under controlled feature-map energies. It does not formally connect to generation quality metrics, and empirical validation (Section 4) demonstrates the practical sufficiency of the retained subspace for video synthesis tasks.

Proof. Because $P_{\le r}$ selects an integer number of complete 2×2 RoPE blocks, the subspaces

$$U_{\le r} = \text{range}(P_{\le r}), \qquad U_{>r} = \text{range}(P_{>r})$$

are invariant under $\mathbf{R}(p)$. Therefore,

$$P_{\le r}\mathbf{R}(p)P_{>r} = 0, \qquad P_{>r}\mathbf{R}(p)P_{\le r} = 0.$$

In other words, $\mathbf{R}(p)$ preserves the split between the retained rotary subspace and the discarded rotary subspace.

Now decompose the features as

$$\Phi(q_p) = \Phi_{\le r}(q_p) + \Phi_{>r}(q_p), \qquad \Psi(k_q) = \Psi_{\le r}(k_q) + \Psi_{>r}(k_q).$$

Then

$$\begin{aligned} s_{\text{full}}(p,q) &= \left(\Phi_{\le r}(q_p) + \Phi_{>r}(q_p)\right)^\top \mathbf{R}(q-p)\left(\Psi_{\le r}(k_q) + \Psi_{>r}(k_q)\right) \\ &= \Phi_{\le r}(q_p)^\top \mathbf{R}(q-p)\Psi_{\le r}(k_q) + \Phi_{\le r}(q_p)^\top \mathbf{R}(q-p)\Psi_{>r}(k_q) \\ &\quad + \Phi_{>r}(q_p)^\top \mathbf{R}(q-p)\Psi_{\le r}(k_q) + \Phi_{>r}(q_p)^\top \mathbf{R}(q-p)\Psi_{>r}(k_q). \end{aligned}$$

By the invariance of $U_{\le r}$ and $U_{>r}$, the two cross terms vanish:

$$\Phi_{\le r}(q_p)^\top \mathbf{R}(q-p)\Psi_{>r}(k_q) = 0, \qquad \Phi_{>r}(q_p)^\top \mathbf{R}(q-p)\Psi_{\le r}(k_q) = 0.$$

Hence

$$s_{\text{full}}(p,q) = \Phi_{\le r}(q_p)^\top \mathbf{R}(q-p)\Psi_{\le r}(k_q) + \Phi_{>r}(q_p)^\top \mathbf{R}(q-p)\Psi_{>r}(k_q).$$

On the first term, $\Phi_{\le r}$ and $\Psi_{\le r}$ live only on the first r coordinates, so

$$\Phi_{\le r}(q_p)^\top \mathbf{R}(q-p)\Psi_{\le r}(k_q) = \Phi_{\le r}(q_p)^\top \mathbf{R}_r(q-p)\Psi_{\le r}(k_q) = s_r(p,q).$$

Therefore,

$$s_{\text{full}}(p,q) - s_r(p,q) = \Phi_{>r}(q_p)^\top \mathbf{R}(q-p)\Psi_{>r}(k_q).$$

Since $\mathbf{R}(q-p)$ is orthogonal,

$$|s_{\text{full}}(p,q) - s_r(p,q)| = \left|\Phi_{>r}(q_p)^\top \mathbf{R}(q-p)\Psi_{>r}(k_q)\right| .$$

By Cauchy–Schwarz,

$$|s_{\text{full}}(p,q) - s_r(p,q)| \le \|\Phi_{>r}(q_p)\|_2 \, \|\mathbf{R}(q-p)\Psi_{>r}(k_q)\|_2 = \|\Phi_{>r}(q_p)\|_2 \, \|\Psi_{>r}(k_q)\|_2 .$$

The last equality uses orthogonality of $\mathbf{R}(q-p)$.

Finally, under Assumption 1, the right-hand side is bounded by $\epsilon_\Phi(r)\epsilon_\Psi(r)$, which gives the stated bound. □

C.3 Obstruction for Inside-Activation Rotary Factorization

The following insight explains why applying RoPE before the pointwise nonlinearity is structurally unfavorable: a pointwise nonlinearity does not commute with a rotation, so absolute position can become entangled with content before global aggregation.

Proposition 3 (No exact relative factorization under inside-activation SiLU)**.** *Let $r \geq 2$, and let $\sigma = \mathrm{SiLU}$ be applied coordinatewise. Define the inside-activation low-rank features*

$$\tilde{q}'_p = \sigma\big(\mathbf{R}_r(p) P_q q^{\mathrm{n}}_p\big), \qquad \tilde{k}'_q = \sigma\big(\mathbf{R}_r(q) P_k k^{\mathrm{n}}_q\big).$$

Then, in the idealized continuous-position setting, there do not exist functions g, h such that for all $p, q \in \mathbb{R}^3$ and all content vectors $q^{\mathrm{n}}_p, k^{\mathrm{n}}_q \in \mathbb{R}^{d_h}$,

$$(\tilde{q}'_p)^\top \tilde{k}'_q = g(q^{\mathrm{n}}_p, k^{\mathrm{n}}_q)\, h(q - p).$$

In particular, if such a factorization existed, the global context

$$C' = \sum_q \tilde{k}'_q v_q^\top$$

would be shift-equivariant (with the readout being shift-invariant); the proposition shows that this is impossible.

Proof. We prove this by contradiction.

Assume that such a factorization exists:

$$\sigma\big(\mathbf{R}_r(p)x\big)^\top \sigma\big(\mathbf{R}_r(q)y\big) = g(x, y)\, h(q - p) \quad \forall p, q \in \mathbb{R}^3, \quad \forall x, y \in \mathbb{R}^r.$$

Step 1: Norm condition from the factorization. Set $q = p$ and $y = x$. Then the left-hand side becomes

$$\sigma\big(\mathbf{R}_r(p)x\big)^\top \sigma\big(\mathbf{R}_r(p)x\big) = \big\|\sigma\big(\mathbf{R}_r(p)x\big)\big\|_2^2,$$

while the right-hand side becomes

$$g(x, x)\, h(0),$$

which is independent of p. Hence for every fixed x,

$$\big\|\sigma\big(\mathbf{R}_r(p)x\big)\big\|_2^2$$

is constant in p.

Step 2: Restrict to a single 2D rotary block. Since $\mathbf{R}_r(p)$ is block-diagonal, it suffices to consider a single 2×2 rotary block with non-zero frequency. Let t denote the rotation angle of that block. Choose x such that only this block is non-zero, and within that block,

$$x = \begin{pmatrix} a \\ 0 \end{pmatrix}, \qquad a > 0.$$

Since $\sigma(0) = 0$ for SiLU, all other blocks contribute zero to the norm squared.

Thus we obtain

$$\big\|\sigma\big(\mathbf{R}_r(p)x\big)\big\|_2^2 = \sigma(a\cos t)^2 + \sigma(a\sin t)^2.$$

By Step 1, this quantity must be independent of t. Define

$$F(t) = \sigma(a\cos t)^2 + \sigma(a\sin t)^2.$$

Then $F(t)$ is constant in t for every $a > 0$.

Step 3: Compute the second derivative at $t = 0$. Let

$$G(u) = \sigma(u)^2.$$

For $u = a\cos t$ and $v = a\sin t$, we have

$$F(t) = G(u) + G(v).$$

At $t = 0$,

$$u(0) = a, \quad u'(0) = 0, \quad u''(0) = -a,$$

and

$$v(0) = 0, \quad v'(0) = a, \quad v''(0) = 0.$$

Since

$$F''(t) = G''(u)(u')^2 + G'(u)u'' + G''(v)(v')^2 + G'(v)v'',$$

substituting $t = 0$ gives

$$F''(0) = G''(a) \cdot 0 + G'(a) \cdot (-a) + G''(0) \cdot a^2 + G'(0) \cdot 0.$$

Thus

$$F''(0) = -aG'(a) + a^2 G''(0).$$

Step 4: Use the derivatives of SiLU**.** For SiLU,

$$\sigma(0) = 0, \qquad \sigma'(0) = \frac{1}{2}, \qquad \sigma''(0) = \frac{1}{2}.$$

Since $G(u) = \sigma(u)^2$, we have

$$G'(u) = 2\sigma(u)\sigma'(u),$$

and

$$G''(u) = 2\left(\sigma'(u)^2 + \sigma(u)\sigma''(u)\right).$$

In particular,

$$G''(0) = 2\left(\sigma'(0)^2 + \sigma(0)\sigma''(0)\right) = 2\left(\frac{1}{4} + 0\right) = \frac{1}{2}.$$

Therefore,

$$F''(0) = -2a\sigma(a)\sigma'(a) + \frac{a^2}{2}.$$

Step 5: Contradiction from constancy of $F(t)$**.** Since $F(t)$ is constant in t, we must have

$$F''(0) = 0.$$

Hence

$$-2a\sigma(a)\sigma'(a) + \frac{a^2}{2} = 0.$$

For $a > 0$, this implies

$$\sigma(a)\sigma'(a) = \frac{a}{4}.$$

Now observe that

$$\frac{d}{da}\left(\sigma(a)^2\right) = 2\sigma(a)\sigma'(a).$$

Thus

$$\frac{d}{da}\left(\sigma(a)^2\right) = \frac{a}{2}.$$

Integrating from 0 to a and using $\sigma(0) = 0$, we obtain

$$\sigma(a)^2 = \int_0^a \frac{t}{2}\, dt = \frac{a^2}{4}.$$

Therefore, for all $a > 0$,

$$\sigma(a)^2 = \frac{a^2}{4}.$$

In particular, for $a = 1$,

$$\sigma(1)^2 = \frac{1}{4}.$$

However, for SiLU,

$$\sigma(1) = \mathrm{SiLU}(1) = \frac{1}{1 + e^{-1}},$$

and

$$\left(\frac{1}{1 + e^{-1}}\right)^2 \neq \frac{1}{4},$$

because $\frac{1}{1+e^{-1}} \neq \frac{1}{2}$. This is a contradiction.

Therefore, the assumed factorization cannot exist. Hence an inside-activation rotation cannot yield a purely relative-position low-rank kernel. □

D Additional Wan2.2 and VBench 2.0 Results

Table 7 extends the main comparison to Wan2.2-A14B-T2V, evaluated under the official recommended setting with 1280×720 resolution, 81 frames, and 40 denoising steps. Table 8 further evaluates RoLA against the corresponding full-attention baseline on Wan2.1-14B-T2V and Wan2.2-A14B-T2V using VBench (Huang et al., 2024) and VBench-2.0 (Zheng et al., 2025). We report the overall VBench score, together with the VBench-2.0 overall score and its five capability categories: Creativity, Commonsense, Controllability, Human Fidelity, and Physics.

Table 7: Additional quantitative comparison on Wan2.2-T2V benchmarks. The layout mirrors the main quantitative table. All measurements use NVIDIA H100 80G GPUs, and latency refers to the end-to-end runtime of the DiT denoising component (excluding text encoding and VAE stages). For the Wan2.2 MoE architecture, this latency measures the denoising computation only and does not include the high-noise/low-noise expert switching time.

Model	Method	VBench Quality					Efficiency	
		SC↑	MS↑	BC↑	IQ↑	AQ↑	Latency↓	Sparsity↑
Wan2.2-A14B-T2V	Full	0.9701	0.9866	0.9629	0.7102	0.6518	1584.6s	0%
	VSA	0.9455	0.9754	0.9497	0.6522	0.5927	986.8s	90%
	SLA	0.9572	0.9812	0.9578	0.6688	0.6182	883.9s	90%
	RoPeSLR	0.9621	0.9850	0.9533	0.6702	0.6193	881.4s	90%
	RoLA (Ours)	0.9645	0.9856	0.9589	0.6805	0.6305	**548s**	90%

Table 8: Comparison of generation quality on VBench and VBench-2.0 for Wan2.1-14B-T2V and Wan2.2-A14B-T2V. Both models are evaluated at 1280×720 with 81 frames on NVIDIA H100 80G GPUs, using their official recommended denoising settings.

Model	Method	VBench	VBench-2.0					
		Total↑	Total↑	Creat.↑	Common.↑	Control.↑	Human Fid.↑	Physics↑
Wan2.1-14B-T2V	Full	83.69	60.20	55.25	63.98	37.32	81.60	62.84
	RoLA (Ours)	83.17	58.23	55.23	58.99	36.23	80.98	59.72
Wan2.2-A14B-T2V	Full	84.21	60.36	55.60	64.50	37.40	81.30	63.00
	RoLA (Ours)	83.52	58.66	55.53	59.21	37.25	81.22	60.11

E Additional Ablation Results

The following experiments are designed to address two potential confounds in the main comparison. First, we fix the sparse attention engine, routing pattern, and target sparsity while changing only the global branch. Second, we evaluate RoLA at multiple sparsity levels. Both tables report the five VBench dimensions used in the main experiment.

E.1 Peak Memory Profiling

Table 12 reports peak GPU memory consumption during inference on Wan2.1-14B at 720p resolution (81 frames). Dense attention without offloading exceeds the 32 GB memory capacity of a single RTX 5090 and results in out-of-memory (OOM) errors. The baseline enables CPU block offloading to fit the model, at the cost of additional host-device transfer overhead. RoLA eliminates the need for offloading entirely, fitting comfortably within on-device memory while achieving $2.2\times$ memory reduction compared to the offloaded dense baseline. This memory efficiency enables deployment on more accessible hardware and allows larger batch sizes or higher resolutions within the same memory budget.

Table 9: Full-attention fine-tuning control on Wan2.1-1.3B at 480p. The first row copies the pre-trained Full Attention result from Table 4; the second row keeps dense attention and applies the same single-stage fine-tuning protocol, adaptation data, batch size, and optimization steps as RoLA.

Setting	SC↑	MS↑	BC↑	IQ↑	AQ↑
Full Attention	0.9403	0.9557	0.9498	0.6510	0.6088
Full Attention + matched fine-tuning	0.9409	0.9556	0.9495	0.6508	0.6084

Table 10: Controlled global-branch ablation on Wan2.1-1.3B at 90% sparsity. The sparse engine and routing pattern are fixed across rows; only the global branch is changed.

Global branch	SC↑	MS↑	BC↑	IQ↑	AQ↑
None (sparse only)	0.8480	0.8912	0.9130	0.4939	0.5281
No RoPE	0.9301	0.9253	0.9241	0.5817	0.5982
RoPE inside activation	0.9303	0.9257	0.9339	0.6027	0.6005
RoPE outside (ours)	0.9521	0.9812	0.9490	0.6492	0.6068

F Training and Model Hyperparameters

F.1 Why Single-Stage Fine-Tuning Replaces the Prior Two-Stage Recipe

The low-rank Fourier compensator (Liu et al., 2026b) used a two-stage post-training procedure because its global module is not a genuine linear-attention branch: it first learns a coordinate-MLP surrogate together with an absolute positional module, and only then co-adapts that surrogate with the full backbone. RoLA changes this optimization picture. Our added parameters

$$\Theta_{\text{new}} = \{P_q, P_k, W_g, b_g\}$$

sit directly on the final low-rank attention path, and their outputs are fused with the sparse branch through the same forward computation used at inference.

This architecture makes a single joint stage the natural default for three reasons. First, the branch is initialized near the sparse baseline because the gate starts almost closed ($W_g = 0$, $b_g < 0$), so early optimization does not abruptly perturb the pre-trained model. Second, the projections P_q, P_k, the gate, and the backbone attention statistics must co-adapt under the final diffusion objective; training them separately would introduce an avoidable stage mismatch between branch fitting and whole-model behavior. Third, unlike the absolute-PE variant of the low-rank Fourier compensator (Liu et al., 2026b), RoLA does not add a separate positional module whose standalone stabilization motivates a warm-up stage.

We therefore adopt a single-stage full fine-tuning recipe for RoLA. This choice should be read as an architecture-matched training strategy rather than a blanket claim that two-stage optimization is inferior for all sparse–low-rank designs. To keep this distinction explicit, we report a controlled single-stage vs. two-stage ablation in Section F.2; Table 13 reports the corresponding comparison under the same VBench setting.

F.2 Training-Strategy Ablation

Table 13 isolates the optimization question from the architectural one. This comparison keeps the backbone, sparsity, data, optimizer, and total update budget fixed, and changes only the post-training schedule: *single-stage* means the joint optimization used by RoLA, whereas *two-stage* follows the corresponding Fourier-compensator warm-up-then-joint recipe. We evaluate both strategies under the same VBench setting to compare their downstream generation quality.

Table 14 lists the main hyperparameters used for the single-stage full fine-tuning (Section 3.5). The newly introduced parameters $\Theta_{\text{new}} = \{P_q, P_k, W_g, b_g\}$ use a higher learning rate than the backbone, since they are trained from initialization while the backbone is only lightly adapted. This stage is deliberately lightweight: RoLA adapts an already pre-trained video DiT rather than learning video generation from scratch. The rotary pairing and frequency schedule are directly inherited from the pre-trained backbone and fixed by the architecture, so the new projections only need to adapt the pre-trained query/key features to the low-rank feature space. In addition, the gate is initialized to

Table 11: Sparsity sweep for RoLA on Wan2.1-1.3B at 480p.

Sparsity	SC↑	MS↑	BC↑	IQ↑	AQ↑
80%	0.9528	0.9820	0.9495	0.6500	0.6072
90% (default)	0.9521	0.9812	0.9490	0.6492	0.6068
95%	0.9503	0.9785	0.9401	0.6485	0.5929

Table 12: Peak GPU memory consumption on Wan2.1-14B (720p, 81 frames) measured on NVIDIA RTX 5090 (32 GB). Both configurations use CPU block offloading to fit within the 32 GB constraint; the reported values reflect on-device GPU memory usage. Dense attention requires 20.43 GiB on-device, while RoLA achieves 2.2× reduction to 9.41 GiB.

Configuration	Peak Memory	Notes
Dense (no offload)	—	OOM (exceeds 32 GB)
Dense (with CPU offload)	20.43 GiB	Baseline (with offloading)
RoLA (ours, with CPU offload)	**9.41 GiB**	**2.2× reduction vs Dense**

a small value, allowing the new branch to be introduced gradually. In our setting, 2,000 training samples over four epochs are sufficient for effective adaptation.

G Computational Complexity Analysis

The rotary-positioned low-rank branch retains the $\mathcal{O}(L)$ scaling of linear attention: the global context

$$C = \tilde{k}^\top V \in \mathbb{R}^{r \times d_h}$$

is formed once at cost $\mathcal{O}(Lrd_h)$ and reused across all queries, and the rank-truncated rotation adds only $\mathcal{O}(Lr)$ elementwise operations. Compared with the absolute-PE variant of the low-rank Fourier compensator in Liu et al. (2026b), which also has $\mathcal{O}(Lrd_h)$ complexity but carries extra learnable positional parameters, RoLA introduces zero additional positional parameters by reusing the backbone rotary tensors. For a fixed sparsity ratio and low-rank width, the asymptotic inference cost of the full module is dominated by the sparse branch, while the low-rank branch contributes only a linear overhead term.

We give a detailed FLOPs analysis of the RoLA attention module during the forward pass, establishing two claims: (i) the rotary low-rank branch retains linear complexity in sequence length, with its cost controlled by the bottleneck width; and (ii) relative to pure sparse attention, the overhead of the low-rank branch is asymptotically negligible for long sequences under a fixed sparsity ratio and low-rank width.

Notation. Let B be the batch size, L the sequence length, H the number of heads, d_h the per-head dimension ($d_{\text{model}} = Hd_h$), S the sparsity ratio (active fraction $1 - S$), and r the low-rank bottleneck. We report dominant matmul FLOPs throughout. A matmul $\mathbb{R}^{M\times N}\cdot\mathbb{R}^{N\times P}$ costs $2MNP$ FLOPs; lower-order operations, including Softmax, RoPE, RMSNorm, sigmoid, and gated addition, are omitted from the FLOPs count.

Cost anatomy. The module decomposes into three parts.

(1) Sparse branch. With block sparsity S, each query attends to $L(1 - S)$ keys; the $QK^\top$ scores and $\mathrm{Softmax}(\cdot)V$ aggregation give

$$C_{\text{sparse}} = 4\,BHL^2(1 - S)\,d_h. \tag{6}$$

(2) Low-rank branch. Down-projection $Q^{\text{n}}P_q^\top$ (and the key side) costs $2BLd_hr$ each; the context $C = \tilde{k}^\top V$ costs $2BLrd_h$; the readout $\tilde{q}C$ costs $2BLrd_h$. Summing over H heads,

$$C_{\text{lowrank}} = 8\,BHLd_h\,r. \tag{7}$$

(3) Gating and fusion. The scalar-gate projection XW_g costs $2BHLd_h$, i.e.,

$$C_{\text{fusion}} = 2BHLd_h. \tag{8}$$

Table 13: Controlled ablation of post-training strategy.

Strategy	SC↑	MS↑	BC↑	IQ↑	AQ↑
Single-stage (ours)	0.9629	0.9853	0.9537	0.6783	0.6294
Two-stage (Fourier-compensator style)	0.9631	0.9842	0.9521	0.6778	0.6296

Table 14: Training and model hyperparameters for RoLA on Wan2.1-T2V-14B.

Hyperparameter	Value
Optimizer	AdamW
β_1, β_2	0.9, 0.999
Weight decay	10^{-5}
Precision	BF16
Low-rank dimension r	64
Gate bias init b_0	-1.946
Global batch size	32
Single-stage full fine-tuning	
Objective	Flow-matching
LR schedule	Cosine
Training samples	2000
Training epochs	4
Learning rates	
Backbone	2×10^{-6}
Low-rank projections (P_q, P_k)	5×10^{-5}
Gate projection W_g	3×10^{-5}
Gate bias b_g	5×10^{-5}

The total is therefore

$$C_{\text{RoLA}} = 4BHL^2(1-S)d_h + 8BHLd_h\, r + 2BHLd_h. \tag{9}$$

Cost comparison with sparse-linear frameworks. A standard full-width linear branch with feature map $\phi : \mathbb{R}^{d_h} \to \mathbb{R}^{d_h}$ forms $KV_{\text{ctx}} = \phi(K)^\top V$ ($2BHLd_h^2$) and retrieves $\phi(Q)KV_{\text{ctx}}$ ($2BHLd_h^2$), giving a dominant matmul cost $C_{\text{linear}} = 4BHLd_h^2$. Under the same FLOPs accounting, our low-rank branch has $C_{\text{lowrank}} = 8BHLd_h\, r$, including the query/key down-projections, context aggregation, and readout. Therefore,

$$\frac{C_{\text{lowrank}}}{C_{\text{linear}}} = \frac{8BHLd_h\, r}{4BHLd_h^2} = \frac{2r}{d_h}. \tag{10}$$

Thus, the low-rank branch is cheaper than a full-width linear branch when $r < d_h/2$, has the same dominant matmul FLOPs when $r = d_h/2$, and remains linear in sequence length in either case. Under our default setting $r = 64$ and $d_h = 128$, the two branches have equal leading-order matmul FLOPs; the advantage of RoLA therefore lies not in reducing this leading-order cost, but in preserving relative-position structure within a compressed rotary feature space.

Asymptotically negligible overhead over pure sparse attention. The extra counted FLOPs relative to the pure sparse baseline are

$$\eta = \frac{C_{\text{lowrank}} + C_{\text{fusion}}}{C_{\text{sparse}}} = \frac{8BHLd_h\, r + 2BHLd_h}{4BHL^2(1-S)d_h} = \frac{2r + 0.5}{L(1-S)}. \tag{11}$$

For the fixed sparsity ratio $S < 1$ and fixed low-rank width r used in our experiments, the number of active keys per query, $L(1-S)$, grows linearly with the sequence length L. Therefore,

$$\eta = \mathcal{O}(L^{-1}), \qquad \lim_{L\to\infty} \eta = 0. \tag{12}$$

Thus, under the fixed-sparsity regime considered in this work, the additional FLOPs introduced by the low-rank branch and gated fusion become asymptotically negligible relative to the sparse-attention computation as the video sequence length increases. In other words, RoLA adds only a linear-cost global branch on top of the sparse attention module, while the dominant sparse-attention cost grows quadratically with L for a fixed sparsity ratio.

H Additional Qualitative Comparisons

The following figure shows representative video samples. RoLA preserves better temporal coherence and scene layout under high sparsity compared with competing sparse-global methods.

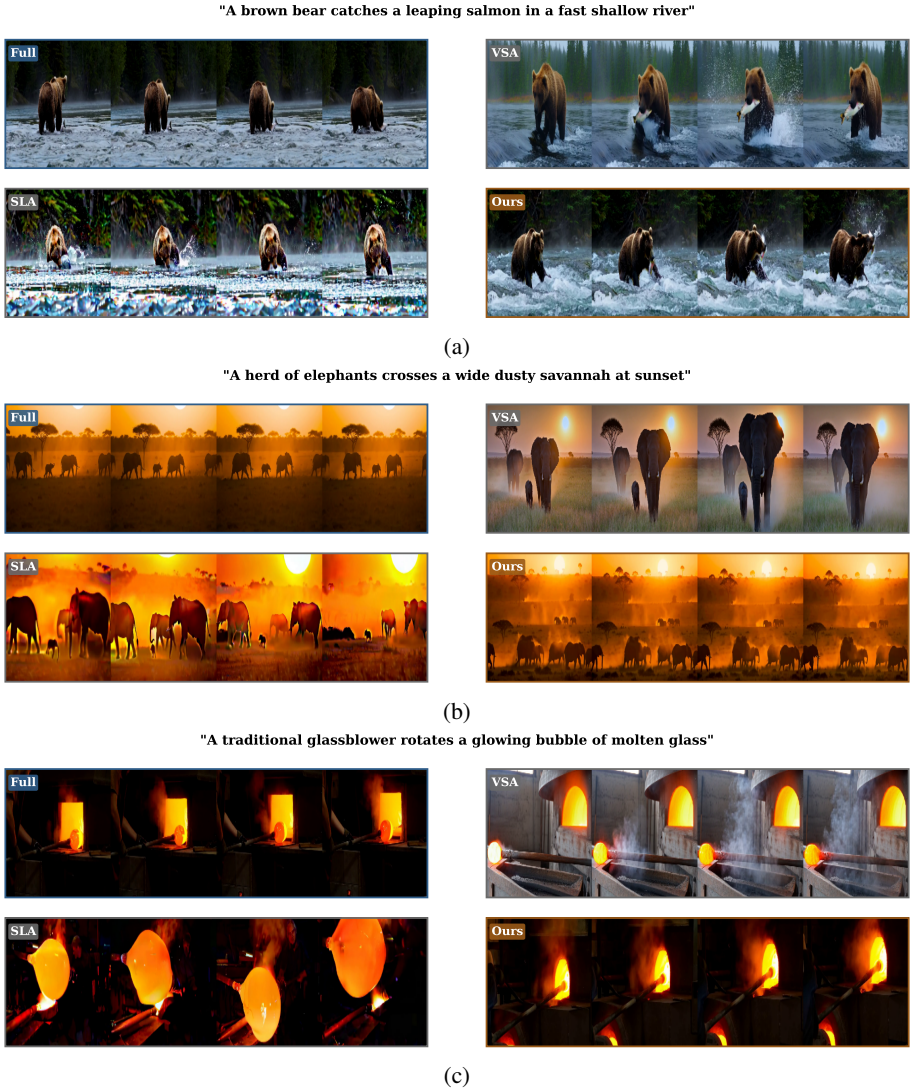
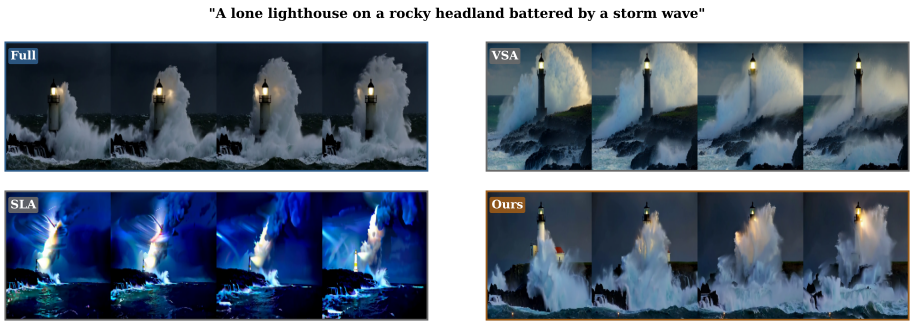


Figure 6: Qualitative comparison on five video prompts. (a) Bear, (b) Elephants, (c) Glassblower, (d) Lighthouse, (e) Tuscany.



(d)



(e)

Figure 6: Qualitative comparison on five video prompts (continued). (d) Lighthouse, (e) Tuscany.