BRAID: Band-Gate Routed Attention with Interleaved Detail for Efficient Video Diffusion Transformers

Anonymous submission

Abstract

Abstract. Exact self-attention is the computational bottleneck of video diffusion transformers: its cost grows quadratically with the full spatiotemporal token sequence, while every other block in the network scales linearly. We present Band-Gate Routed Attention with Interleaved Detail (BRAID), a sparse-linear attention module that replaces dense self-attention with two complementary paths: a query-adaptive block-sparse branch that evaluates exact softmax on selected key blocks under a fixed budget; and a subband-gated linear complement that carries global context across Haar wavelet subbands through learned per-head gains. A structural sink reservation stabilizes the sparse normalizer, and a denoise-progress-aware schedule reallocates one average block budget across the sampling trajectory. BRAID is defined at the attention-module level and is not tied to a particular backbone. We establish a unified cost decomposition across Dense as an upper-bound reference, SLA as the direct sparse predecessor, and BRAID, keeping this arithmetic analysis separate from matched measurement. On the full evaluation set, BRAID improves prompt following over its direct sparse predecessor SLA while remaining within a narrow margin of the exact-attention reference; on the reference hardware, its average per-step wall-clock time is 47.1% lower than that of the exact-attention step of the same network.

Keywords: *video diffusion transformers; sparse attention; linear attention; block-sparse routing; wavelet subbands*

1. Introduction

Video diffusion makes attention a geometry problem before it becomes a kernel problem. A video DiT attends over the entire spatiotemporal token sequence at once, so its self-attention cost grows quadratically with a token count that no windowing removes, while every other term in the network scales linearly. At the fixed evaluation geometry of this manuscript, dense self-attention is about 73% of the per-step arithmetic of the exact-attention network (Table 1): attention is not one bottleneck among several but the dominant term of the dominant computation. Exact-attention implementations [3, 4] make such sequences memory-feasible, but they do not remove the quadratic arithmetic. BRAID is aimed squarely at that term: it replaces dense self-attention with a sparse-linear module that is defined at the attention-module level and is not tied to a particular backbone. Figure 1 anchors the output that module produces, shown without comparison; the arm-by-arm visual and scored comparisons follow in Section 4.3.



Figure 1. BRAID output under one seed, sampler, and block budget, shown as full clip strips without comparison: seed 1, 50 sampling steps at CFG 5.0, block budget 0.15; the generating prompts are quoted in full below the figure.

Top row: A golden retriever runs across a grassy field and leaps into a shallow stream, water splashing around it.

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.

Middle row: A chef flips a pan over a high flame, tossing vegetables that arc briefly into the air before landing back.

Bottom row: A hummingbird hovers near a red flower, darts in to sip nectar, then flits away to the next bloom.

Table 1. Shared-scope analytical per-step FLOP ledger for Wan2.1-T2V-1.3B at 480p (832×480), 81 frames: L = 32,760 tokens, d = 128, H = 12 heads, N = 30 layers.

Component, one forward step	Dense	SLA	BRAID	Accounting basis
Self-attention path, 30 layers	2.0e14	2.9e13	2.9e13	dense $4L^2\,dH$; selected $4L(kB)dH$
Linear complement, 30 layers	—	1.2e12	1.2e12	SLA global summary; BRAID subband summaries
Pooled routing, 30 layers	—	2.7e10	2.7e10	block means + pooled scores
FFN + modulation, 30 layers	5.4e13	5.4e13	5.4e13	shared ledger term
q, k, v, o projections, 30 layers	1.9e13	1.9e13	1.9e13	shared ledger term
Analytical total	2.7e14	1.0e14	1.0e14	same shared scope; 38.2% and 38.2% of the Dense total for SLA and BRAID respectively

Note. Analytical FLOPs only; this table does not report kernel time, step latency, or any wall-clock measurement. SLA rounds its nominal $\rho = 0.15$ *to* $\mathrm{int}(0.15 \times \mathrm{ceil}(32760/64)) = 76$ *of 512 key blocks (effective block density 0.1484375). Dense counts* $4L^2\,dH$ *and selected branches count* $4L(kB)dH$*. SLA linear counts* $k^\top v$*,* $q(k^\top v)$*,* proj_l*, and two normalizer vectors; SLA routing counts block means and the pooled-score matrix product. The BRAID subband complement counts the same accumulation terms as SLA linear, because one Haar level places* $L/4$ *tokens in each of four subbands and the four per-subband accumulations sum to the flat count; BRAID routing counts the same block means and pooled-score product as SLA routing. Elementwise feature maps, wavelet adds, normalization, Top-k sorting, scatters, and kernel instructions are outside this analytical ledger.*

BRAID is built around a division of labor. A sink-reserved query-adaptive block-sparse branch preserves exact softmax interactions on a budget-accounted support. A subband-gated linear branch carries global context without forcing coarse layout and fine detail to share one flat summary. A denoising-progress schedule then reallocates one average block budget over the sampling trajectory.

Measured on the evaluation set of Section 4, that division of labor moves the sparse operating point: prompt following improves over SLA, the direct sparse predecessor, while the residual gap to the exact-attention reference is confined to prompt following, aesthetic quality, and dynamic degree, and the step time drops by a large fraction. The contribution of the module is therefore the trade-off it offers, not a claim that a sparse construction reproduces the reference.

Our contributions are twofold. First, we introduce BRAID, a two-path sparse-linear attention module for video diffusion transformers that combines budgeted exact block-sparse support and subband-gated linear global summaries, and show that the subband split costs the same arithmetic as one flat summary while keeping coarse layout and fine detail from blurring into each other. Second, we provide a shared-scope analytical decomposition against the upper-bound reference Dense and its direct predecessor SLA at fixed geometry, together with a matched evaluation protocol that separates arithmetic accounting from quality and latency evidence.

2. Related Work

2.1 Video diffusion transformers

Diffusion transformers [11] replace the convolutional U-Net with a transformer over latent tokens, and video systems extend that token grid along time: the open text-to-video backbone used throughout this study [12] flattens a spatiotemporal latent into one sequence and attends over it, as do the other members of this family. The sequence length grows with the product of frame count and frame area, so self-attention grows quadratically in a quantity that a user raises by asking for a longer or larger clip, and it comes to dominate the per-step cost while the FFN and projection terms stay linear. This is what makes the attention budget, rather than the parameter count, the design variable in this family, and it is the regime the present work addresses.

2.2 Exact and sparse attention

FlashAttention [3, 4] reduces the memory cost of exact attention through IO-aware tiling, but the dense attention arithmetic remains quadratic. Sparse Transformers [2] established that structured attention patterns can reduce this cost, and the designs that followed differ mainly in where the pattern comes from. Static layouts combine local windows with a few global positions [1, 14], hashing selects candidate keys from the content itself [8], and recent work selects the pattern per input, either at inference time from observed attention structure [6] or by training the model with the selection in the loop [13] so that the sparse operator is the one the weights were fitted to. SLA [15] is a published sparse-linear video attention of this kind; we use it as the sparse-linear baseline in our unified cost decomposition. BRAID targets the same arithmetic regime with an independent construction: its sparse branch is a standalone TileLang kernel that evaluates exact softmax over dynamically selected blocks, and the routing, budget schedule, and complements are specified in Section 3 without recourse to the SLA design.

The contrast with the SLA baseline can be stated exactly, because both methods route per query block through pooled scores; they differ in what surrounds that shared operation. SLA centers the keys at token level before pooling, scores by pooled dot product, applies one flat budget ratio across layers and steps, reserves no sink block, and attaches a single global linear summary to the sparse branch. BRAID centers the pooled key descriptors themselves in fp32 and L2-normalizes both sides, so it scores by cosine, protects the leading block at every budget, moves the budget along the denoising trajectory at a fixed mean, and attaches the subband-gated complement of Section 3.4.

2.3 Linear complements

Linear attention [7] replaces the full attention matrix with feature-map summaries that scale linearly in token count, and wavelet-space attention [16] shows that frequency-decomposed representations can structure attention computation. The linear branches of BRAID occupy this space without replacing the selected softmax entries: the subband complement (Section 3.4) decomposes q, k, and v with one level of orthonormal Haar wavelets, runs an independent linear attention per subband, and gates the four contributions with learned per-head gains before reconstruction.

3. Method

Throughout, L denotes the spatiotemporal token count of one video latent, d the per-head dimension, H the number of heads, and N the number of transformer layers re-using the module. For one batch element and head, the dense reference attention that BRAID replaces is

$$O^* = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{1}$$

with Q, K, V the query, key, and value projections. Every sparse or linear construction below is defined by its relation to this expression, which remains available as the measurement oracle.

BRAID replaces dense self-attention with two cooperating paths under a per-step block budget that every layer obeys, merged through initialization-safe parameters so that the added branches are inactive at the start of fine-tuning.

The design is organized around one falsifiable claim. Hypothesis 1 (scale-structured compensation): the mismatch of a flat linear summary against the exact reference is structured by spatial scale, and per-head subband gains supply the matching degrees of freedom; Figure 4. The routing and budget construction of the sparse branch (Sections 3.1-3.3 and 3.5) is argued on its own terms in those sections rather than set up as a second hypothesis.

We write the method on top of the flow-matching formulation used by the base model [12, 10]: the noised latent is $x_\sigma = (1 - \sigma)\, x_0 + \sigma\, \varepsilon$ and the network predicts the velocity $v = \varepsilon - x_0$. Each layer routes from its own pooled descriptors; the one quantity fixed once per denoising step is the budget ratio, which every layer then obeys.

3.1 Two-channel base: block-sparse attention with a linear complement

The token sequence is partitioned into K key blocks of width $B = 64$ and as many query blocks, the final block possibly ragged. The width balances granularity against overhead: narrow enough that routing can drop a locally irrelevant region without discarding the tokens around it, coarse enough that the pooled-descriptor matrix and the

top-k bookkeeping stay small. Block descriptors are the per-block means of the query and key projections, written with an overbar, and routing scores every query block against every key block by the cosine between the query descriptor and the mean-centered key descriptor (Section 3.3):

$$s_{ij} = \frac{\underline{q}_i}{\left(\left\|\underline{q}_i\right\|\right)} \cdot \frac{\underline{k}_j - m}{\left(\left\|\underline{k}_j - m\right\|\right)}, \quad \mathrm{T}(i) = \mathrm{TopK}_j\{s_{ij}\}, \quad k = \lfloor \rho K \rfloor \tag{2}$$

where TopK returns the k largest scores and $k = \lfloor \rho\, K \rfloor$ with ρ the budget ratio, evaluated per batch element, head, and query block. The sparse branch then evaluates exact softmax attention restricted to the selected support,

$$o_s(i) = \sum_{j \in \mathrm{T}(i)} \mathrm{softmax}_j\left(\frac{q_i {k_j}^T}{\sqrt{d}}\right) v_j \tag{3}$$

where the normalizer runs over the selected blocks only; the kernel therefore implements a genuine attention distribution on $\mathrm{T}(i)$, not a masked dense attention. The linear branch accumulates a global summary,

$$S = \sum_n \left(\varphi(k_n)\right)^T v_n, \quad z = \sum_n \left(\varphi(k_n)\right), \quad o_l(i) = \frac{\left(\varphi(q_i)\right) S}{\left(\varphi(q_i)\right) z + \varepsilon} \tag{4}$$

with φ a per-token feature map (softmax over the head dimension in our runs), and the two branches merge as $o = o_s + W_l\, o_l$ through a zero-initialized projection W_l.

Proposition 1 (initialization identity and gradient opening). At initialization $W_l = 0$, so the module output equals the block-sparse branch exactly: $o(i) = o_s(i)$ for every query i, and the module is the pretrained dense attention of Eq. (1) evaluated on the selected support. Moreover the output responds to the added projection at first order: the differential of the merge along the linear branch is $(\mathrm{d}W_l)\, o_l$, which is nonzero for generic inputs because S and z accumulate over all tokens.

Proof. Substituting $W_l = 0$ in the merge removes the complement term and gives the identity; the derivative of o with respect to W_l at the initial point is o_l itself, since the merge is linear in $W_l\, o_l$. ■

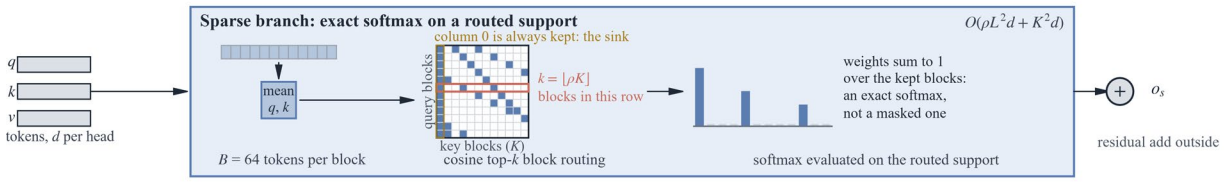

Figure 2. Construction of the block-sparse branch (Sections 3.1-3.3): the layer input pooled into block descriptors, the per-query-block routing matrix with the reserved sink column kept in every row, and the softmax evaluated over the kept blocks alone. Glyphs state the construction, not a measurement.

3.2 Sink reservation

One reservation corrects an instability that is structural rather than semantic. Block 0 is forced into every support set inside the same budget: the reserved block is carved out of the k slots and displaces the lowest-ranked content block, so the support stays at exactly k blocks and the reservation costs no extra arithmetic. Trained softmax attention puts a disproportionate share of its mass on a few positions rather than spreading it, and in the base model used here that mass sits on the leading tokens, so a row whose content ranking drops that block has an uncalibrated normalizer, which surfaced as flickering artifacts in development runs without the reservation. The first block is therefore a natural anchor: reserving it is intended to keep the normalizer calibrated independently of content. The reservation is sample-independent and therefore identical across guidance branches, so it cannot introduce a guidance-dependent asymmetry.

3.3 Pooled key centering and directional scoring

Routing subtracts the global key mean m from the pooled descriptors rather than from the tokens, and this reordering is exact:

$$\text{mean}_b(k - m) = \text{mean}_b(k) - m, \quad m = \frac{1}{L}\sum_b |b| \cdot \text{mean}_b(k) \tag{5}$$

Proposition 2 (pooled centering is exact and count-consistent). For any partition of the keys into blocks, $\text{mean}_b(k - m) = \text{mean}_b(k) - m$ holds identically, and the global mean is recovered exactly from the block means as the count-weighted average $m = (1/L)\sum_b |b| \cdot \text{mean}_b(k)$, the ragged tail block entering with its true token count.

Proof. Both identities are linearity of the mean: subtracting a constant commutes with averaging, and the count-weighted sum $\sum_b |b| \cdot \text{mean}_b(k)$ collects every key exactly once, so it equals $L \cdot m$. ■

The identity is what makes the reordering useful: the centered descriptors can be formed in fp32 from the pooled values instead of materializing a centered copy of the full key tensor, so the reordered computation touches only the K pooled descriptors rather than L keys per layer, and it avoids rounding the small centered values to bf16 before pooling, precisely where relative rounding error is largest. On top of the centered descriptors we score by cosine rather than dot product in every configuration reported here, ranking blocks by direction instead of norm; this matters at small budgets, where a high-norm but irrelevant block would otherwise win a slot.

3.4 Subband-gated linear complement

The flat linear branch of Eq. (4) mixes all scales into one d-by-d summary, so coarse layout and fine detail compete for the same capacity, and a fixed cross-scale merge cannot express a distinct decision for each head and subband. We instead separate scales before accumulation: each of q, k, v is reshaped to the (f, h, w) token grid and decomposed by one level of orthonormal Haar wavelets over (h, w), giving four subbands $s \in \{\text{LL}, \text{LH}, \text{HL}, \text{HH}\}$ at half resolution; inside each subband an independent linear attention runs with its own summary and normalizer, its output is scaled by a learnable per-head gain $g_{h,s}$, and the inverse Haar transform reconstructs,

$$y_s = \frac{g_{h,s} \cdot \varphi(q_s) S_s}{\varphi(q_s) z_s + \varepsilon}, \quad o_l = \text{IDWT}(y_{LL}, y_{LH}, y_{HL}, y_{HH}) \tag{6}$$

Proposition 3 (lossless, energy-preserving split). One level of orthonormal Haar wavelets satisfies $\text{IDWT}(\text{DWT}(x)) = x$ for every token tensor x, and for even spatial extents Parseval's identity distributes the squared norm exactly over the four subbands:

$$\text{IDWT}\big(\text{DWT}(x)\big) = x, \quad \sum_s \|x_s\|^2 = \|x\|^2 \tag{7}$$

Proof. The Haar transform is an orthogonal change of basis on the spatial axes; orthogonality yields both the reconstruction identity and norm conservation on the transform grid. An odd spatial extent is edge-padded before the transform, so reconstruction still returns x after cropping while the energy split holds on the padded grid, which is why the norm identity is stated for even extents. ■

The split therefore neither loses nor invents information. The only learned freedom distinguishing the subband branch from the flat linear branch of Eq. (4) is the per-head gain vector $g_{h,s}$; the per-subband summaries and normalizers are determined by the input, not learned. The DWT and IDWT are adds and subtracts, $O(L\,d)$, negligible against the $O(L\,d^2)$ accumulation; one Haar level splits the L tokens into four subbands of $L/4$ tokens each, so the four per-subband accumulations sum to the same $O(L\,d^2)$ work as the single flat accumulation and the branch costs what the flat linear branch costs up to those transforms, more than an order of magnitude below the FFN. The gain is the design's band gate: initialized to one, it is the identity at the start of training, and fine-tuning moves it to whatever trust per scale each head warrants, with no hand-picked cutoff and no assumption that the horizontal and vertical detail bands should be treated alike. Reconstruction is where the detail returns interleaved: the inverse transform writes the three separately gated detail bands back to the same token positions as the gated layout band, so each head re-admits its detail at its own strength alongside the layout instead of through one merged summary. This is the interleaved detail the method is named for.

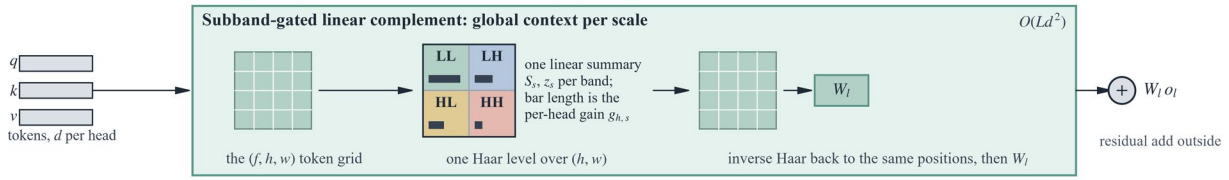

Figure 3. Construction of the subband-gated linear complement (Section 3.4): one Haar level splits the token grid into four bands, each band accumulates one linear summary, the per-head gains gate the bands, and the inverse transform writes them back to the same token positions before the learned projection. Glyphs state the construction, not a measurement.

The recorded internal diagnostic probe in Figure 4 measures this compensation opportunity against per-subband exact softmax attention: its oracle scalar gains and linear-reference error reductions vary across layers and subbands, and the plotted uncertainty is the spread of the per-head least-squares gains inside a subband, the freedom the per-head parameterization supplies.

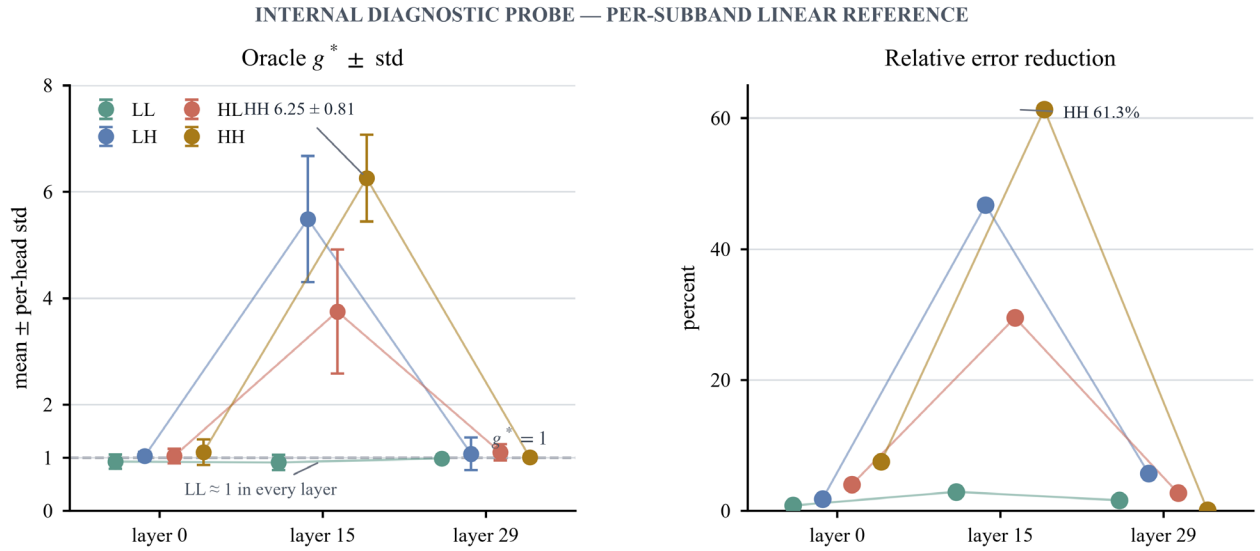

Figure 4. Internal diagnostic probe of the per-subband linear reference, scored against per-subband exact softmax attention. (a) Oracle scalar gain per layer and subband, with the spread of the per-head least-squares gains. (b) The relative error the same gain removes. Both panels describe the linear reference itself and carry no claim about the rendered clip.

3.5 Budget schedule over the sampling trajectory

Not all steps need the same budget. Early steps commit the spatial layout and late steps settle detail, while the middle of the trajectory refines a structure that is already decided. We make the top-k ratio a piecewise-linear function of the flow progress r ($r = 1$ at pure noise, 0 at clean) through three control points (start, mid, end), and we choose the accounting so the comparison against a flat baseline is exact in the mean: r is read off the uniform σ grid, so the mean budget over a run is the integral of the piecewise-linear curve,

$$\rho_{avg} = \frac{\rho_{start} + 2\rho_{mid} + \rho_{end}}{4} \leq \rho_{nominal} \tag{8}$$

with equality whenever the trapezoidal mean $(\text{start} + 2 \cdot \text{mid} + \text{end})/4$ of the control points equals the nominal ratio, the flat schedule being one such case. The default uses the control points (0.19, 0.10, 0.20): both trajectory ends receive a fuller budget while the middle pays for it, and the mean 0.1475 stays below the 0.15 flat budget used for comparison. The mean is never raised to pay for the fuller ends, so the flat baseline remains the accounting counterpart and the comparison is conservative in its favor. The schedule is applied identically in training and inference, and the sink reservation of Section 3.2 reserves block 0 at every budget level. This study

does not isolate the schedule's contribution from the rest of the module: the rendered comparison of Section 4 carries the design as a whole, and the schedule enters it as a fixed setting rather than as a separately tested claim.

3.6 Training objective and initialization

The sparse branch consumes only the selected query–key block pairs under an online softmax and is a standalone TileLang kernel; the routing map is recomputed in every layer from that layer's own pooled descriptors, while the budget ratio it obeys is fixed once per denoising step by the schedule of Section 3.5. BRAID adds a small trainable set on top of the backbone: the linear projection W_l, the self-attention output projection of each block, and the subband gains $g_{h,s}$. All are optimized under the unchanged flow-matching velocity objective of the base model; no auxiliary loss is introduced. By Proposition 1 every added path is exactly inactive at the start of training, so the fine-tuned model begins at a degraded-support version of the pretrained network rather than at an uncontrolled perturbation of it. This initialization discipline also fixes the meaning of an ablation: removing a component defines a separately trained configuration, not a post-hoc switch on an incompatible checkpoint.

3.7 Complexity summary

Table 1 fixes the arithmetic scope: one forward denoising step of the fixed instantiation, whose token count, head dimension, head count, and layer count are recorded in that caption. The dense reference attention of Eq. (1) costs

$$C_{attn}^{dense} = N(2L^2dH + 2L^2dH) = 4NL^2dH \tag{9}$$

the two summands being the score product $QK^\top$ and the value aggregation, each quadratic in L. A block-sparse evaluation over k selected key blocks of width B replaces the key extent L by kB in both products,

$$C_{sel}(k) = 4NL(kB)dH = \frac{kB}{L} \cdot C_{attn}^{dense} \tag{10}$$

so the selected-attention cost is the token-weighted density kB/L of the dense term, tracking the budget ratio ρ up to the single-block rounding of the floor, while every remaining ledger entry is budget-independent. The per-step total of either method is therefore affine in ρ with slope equal to the dense attention term of Eq. (9); this is shown in Table 1 where the two sparse methods differ only in their constant terms.

Memory traffic follows a different accounting, and the two paths sit in different places under it. The exact branch inherits the IO-aware tiling of FlashAttention [3, 4]: its online softmax runs over the selected query-key block pairs without ever materializing a score matrix, so the selected support changes how many tiles are visited, not the resident working set. The subband complement adds only the wavelet temporaries, $O(L\,d)$ per head, alongside summaries that are $O(d^2)$ and independent of L.

4. Experiments

4.1 Evaluation protocol and claim boundaries

Hypothesis 1 of Section 3 is the mechanism claim read from matched records: the subband probe (Figure 4) together with the scored gain on/off pair over the evaluation set. The routing and budget construction is argued on its own terms in Section 3 and judged by the rendered comparison of this section; it is not carried to a separate mechanism verdict here.

The rule that turns those records into a verdict is fixed here, before any of them is reported. The mechanism claim is judged on how well the render answers the prompt it was given: switching the mechanism off must cost more than 0.005 points of prompt-following score over the evaluation set, and a smaller separation in either direction is reported as undecided at the pre-registered margin rather than as a win. Appearance, subject and background stability, and amount of motion are read alongside that score so a gain in one cannot be traded quietly against a loss in another. Pixel agreement with the exact-attention render is deliberately not the axis: it rewards the arm that moves least and would make imitation of that render the goal, which is neither what a viewer sees nor what this design is for.

4.2 Scope of reported evidence

The analytical result in this manuscript is the cost ledger in Table 1, which is arithmetic rather than measurement. The recorded material in Figure 4 is a local diagnostic of the linear reference. Everything else reported below comes from renders of the three arms at one fixed geometry, one sampler, and one fixed trained state per arm; the scored numbers average over 800 prompts and 5 seeds of that fixed setting, and no number is carried over from a different geometry, sampler, or checkpoint. Latency is a wall-clock measurement, not a ledger item: the per-step timings in Section 4.3 were taken on the same hardware at the evaluation geometry for the three arms, and the ledger itself stays arithmetic.

4.3 Results

Every visual panel in this manuscript is shown from a single seed for legibility; the render condition used throughout the visual material is: seed 1, 50 sampler steps at CFG 5.0, nominal block budget 0.15 with control points (0.19, 0.1, 0.2) of trapezoidal mean 0.1475 on the BRAID arm. The scored comparison covers 800 prompts rendered under 5 seeds with the same sampler and block budget on every arm, and the tables below report means over those prompts and seeds.

	Dense	SLA	BRAID
Key frame			
Detail crop			

Figure 5. Matched run material: Dense, SLA, and BRAID key frames and detail crops, every cell rendered under the render condition stated at the head of this section. Prompt-following and appearance scores over the full evaluation set are listed in Table 2. The generating prompt of this panel is quoted in full below the figure. Dense is the upper-bound reference; the three columns are directly comparable because only the attention module differs between them.

The panel renders one long prompt, quoted in full: "A pianist keeps a rolling bass line with her left hand while the right leaps across the high keys, and mid-phrase she reaches over and turns the page without breaking tempo. She sways into the crescendo, then softens to a whisper for the coda, her eyes flicking between the score and her hands. A slow crane shot rises from the keys to her profile. Sheet music and pencil marks fill the warm concert hall light."

Standard benchmark suites often feature short, generic captions; this regime chains four composite performer actions inside a single clause and carries the camera movement in the same sentence, so a render that holds one plausible pose still fails it. On this prompt the BRAID render varies the performer's posture and gaze across the clip, the dense reference holds a single seated reading pose, and the SLA render moves within a visibly narrower envelope. A single long prompt is not a measurement, but it is the regime the sparse construction is asked to keep following once the prompt specifies more than a layout.

Table 2. Prompt-following and appearance scores over the 800-prompt evaluation set (5 seeds), one score per arm per metric, averaged over the prompts and seeds, scored by the standard suite [5]. The prompt-following row is that suite's VQAScore [9] and the axis the mechanism claim of Section 4.1 is read on. Every arm rendered the same prompts under the same 5 seeds and sampler, so a column differs from its neighbour only in the attention module it used.

Metric	Dense	SLA	BRAID
Prompt following (VQAScore)	0.5969	0.5701	0.5878
Aesthetic quality	0.6006	0.5454	0.5591
Subject consistency	0.9329	0.9331	0.9436
Background consistency	0.9633	0.9619	0.9653
Imaging quality	0.7052	0.7106	0.7159
Dynamic degree	0.7100	0.6200	0.6600
Motion smoothness	0.9843	0.9858	0.9859
Temporal stability	0.9750	0.9768	0.9756

Note. Higher is better on every row, and each row has its own scale, so a row is read across the arms and not against another row. A low dynamic degree means a nearly still clip rather than a calm one. Dense is the exact-attention upper-bound reference, not a target to reproduce. The prompt-following row is scored by the suite's VQAScore judge (Qwen2.5-VL-7B over the official v3.1 templates of the t2v metrics suite). The four columns are scored on the identical rendered set.

Table 3. Mechanism ablation on the same axis: BRAID against the separately trained arm with the subband gain mechanism switched off.

Metric	BRAID	BRAID, subband gain off
Prompt following (VQAScore)	0.5878	0.4973
Aesthetic quality	0.5591	0.5464
Subject consistency	0.9436	0.9467
Background consistency	0.9653	0.9655
Imaging quality	0.7159	0.7044
Dynamic degree	0.6600	0.7600
Motion smoothness	0.9859	0.9843
Temporal stability	0.9756	0.9720

Note. The BRAID column and the gain-off column are both measured over the evaluation set.

The scored comparison is read from Table 2. On the prompt-following axis of that table, BRAID reaches 0.5878 against 0.5701 for SLA, a lead of +0.0177 over its sparse-linear predecessor, and the exact-attention reference scores 0.5969 on the same axis, -0.0091 relative to it. The lead over SLA is not bought with a stiller picture: dynamic degree is 0.6600 against 0.6200 for SLA, and motion smoothness is 0.9859 against 0.9858 for SLA, with the exact-attention reference at 0.7100 and 0.9843 on the two rows. Aesthetic quality is 0.5591 against 0.5454 for SLA and 0.6006 for the exact-attention reference. Relative to SLA, BRAID is lower on temporal stability and at parity or higher on the remaining reported rows. Against the exact-attention reference, BRAID trails on prompt following (vqascore), aesthetic quality, and dynamic degree and is at parity or above on every other row; the sparse construction therefore does not reproduce the reference arm-for-arm, and understanding how the sparse support and the subband complement shift the trade-off between prompt saliency and temporal conservatism relative to the dense operating point remains future work.

Wall-clock latency is in Table 4: BRAID's per-step time (1.19 s) is no slower than SLA's (1.25 s), and its average per-step wall-clock time is 47.1% lower than that of the exact-attention step of the same network (2.25 s), equivalently a 1.89× speedup, so the added complement branches cost no step time at this batch size, under the protocol stated in the table's note. Under CFG the sampler runs the network once per guidance branch, so the per-step wall time of every arm approximately doubles while the relative comparison is unchanged.

Table 4. Measured wall-clock companion to the analytical ledger of Table 1: one forward denoising step of the full network for each arm.

Method	One forward step	Share of Dense step time	Analytical FLOP share
Dense	2.25 s	100%	100%
SLA	1.25 s	55.5%	38.2%
BRAID	1.19 s	52.9%	38.2%

Note. Protocol: single NVIDIA H20 GPU, bf16, batch size 1, one forward denoising step of the full network with a single sampling branch (no CFG duplication); reported times are the mean over the 50 denoising steps of the sampler, taken in a separate run that follows a three-step warm-up. The measured share of the Dense step time is bounded by memory traffic and kernel behavior and need not match the analytical FLOP share.

Hypothesis 1 is read from two records. The oracle gains in Figure 4 show per-band mismatch that a learnable gain can correct; what the correction does to the output is the BRAID, subband gain off arm of Table 3, retrained under the same recipe with the subband gains switched off. Switching the gains off drops the prompt-following score from 0.5878 to 0.4973, a gap of 0.0905, well beyond the 0.005-point margin fixed in Section 4.1, so Hypothesis 1 is supported at that margin. The rows that move the other way are small except dynamic degree: subject and background consistency rise by under 0.01 each without the gains, while dynamic degree rises from 0.66 to 0.76; the gain-off clips move more and score far lower on the prompt-following axis. Motion quantity does not order the scores of the main arms: SLA moves least of the three and still scores below BRAID, which itself moves more than SLA. The ablation therefore states the trade-off of the mechanism directly: removing the gains raises dynamic degree from 0.66 to 0.76 while the prompt-following score falls from 0.5878 to 0.4973. The gains do restrict motion, and the extra motion of the gain-off arm coincides with the collapse on the prompt-following axis; separating motion the prompt licenses from motion as such, per prompt, is left to future work.

The overall reading is a trade-off rather than a reproduction claim: within the sparse family the module improves the operating point of its predecessor (Table 2, Table 3), and the residual gap to the exact-attention reference on the rows where it trails is exchanged for the step-time reduction of Table 4. A sparse construction that simply copied the dense operating point would have nothing to offer; this one offers a better sparse operating point at a large fraction of the dense reference's quality.

5. Limitations

The subband branch requires the plain (f, h, w) token layout. Configurations that prepend reference latents change the index arithmetic and are rejected rather than silently routed. The linear complement is a mean-field summary, so it cannot restore a local error made by the sparse branch. The subband decomposition operates on the spatial axes only; extending the Haar transform along the temporal axis to disentangle motion frequencies remains an open direction.

The sink reservation encodes a prior that leading tokens concentrate useful mass; for inputs without persistent objects, the reserved slot may carry little information. Generalization beyond the single backbone and evaluation geometry recorded here remains to be established, as does an isolated comparison of the routing recipe, the sink reservation, or the schedule: the present evidence carries the module as a whole rather than separating the contribution of any one of its parts.

6. Conclusion

BRAID is a division-of-labor sparse-linear attention design for one fixed video diffusion setting: a sink reservation, a budget-accounted top-k schedule, and a subband-gated linear complement, with initialization identities that keep every added path inactive at the start of training. The one claim carried to a verdict, scale-structured compensation, is stated with its metric and matched-record requirement in advance, so it can be refuted by the protocol of Section 4.

The measured case for the module is the trade-off it offers inside the sparse family: on the full evaluation set it improves prompt following over its direct sparse predecessor while keeping the residual gap to the exact-attention reference confined to a few rows of the scored table, and it exchanges that residual gap for a large reduction in per-step wall-clock time. A sparse construction is worth adopting when it moves the sparse operating point rather than

when it reproduces the dense one; this design is offered on that ground, with the verdict stated and the protocol that can refute it kept in the open.

References

[1] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The Long-Document Transformer, arXiv:2004.05150, 2020.

[2] R. Child et al., Generating Long Sequences with Sparse Transformers, 2019 (arXiv:1904.10509).

[3] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, C. Ré, FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness, NeurIPS 2022 (arXiv:2205.14135).

[4] T. Dao, FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning, ICLR 2024 (arXiv:2307.08691).

[5] Z. Huang et al., VBench: Comprehensive Benchmark Suite for Video Generative Models, CVPR 2024 (arXiv:2311.17982).

[6] H. Jiang et al., MInference 1.0: Accelerating Pre-filling for Long-Context LLMs via Dynamic Sparse Attention, arXiv:2407.02490, 2024.

[7] A. Katharopoulos, A. Vyas, N. Pappas, F. Fleuret, Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention, ICML 2020 (arXiv:2006.16236).

[8] N. Kitaev, Ł. Kaiser, A. Levskaya, Reformer: The Efficient Transformer, ICLR 2020 (arXiv:2001.04451).

[9] Z. Lin et al., VQAScore: Evaluating Text-to-Visual Generation with Image-to-Text Generation, ECCV 2024 (arXiv:2404.01291).

[10] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, M. Le, Flow Matching for Generative Modeling, ICLR 2023 (arXiv:2210.02747).

[11] W. Peebles, S. Xie, Scalable Diffusion Models with Transformers, ICCV 2023 (arXiv:2212.09748).

[12] Team Wan, Wan: Open and Advanced Large-Scale Video Generative Models, arXiv:2503.20314, 2025.

[13] J. Yuan et al., DeepSeek-AI, Native Sparse Attention: Hardware-Aligned and Natively Trainable Sparse Attention, arXiv:2502.11089, 2025.

[14] M. Zaheer et al., Big Bird: Transformers for Longer Sequences, NeurIPS 2020 (arXiv:2007.14062).

[15] J. Zhang et al., SLA: Beyond Sparsity in Diffusion Transformers via Fine-Tunable Sparse-Linear Attention, arXiv:2509.24006, 2025.

[16] Y. Zhuang, Z. Wang, F. Tao, J. Shang, WavSpA: Wavelet Space Attention for Boosting Transformers' Long Sequence Learning Ability, arXiv:2210.01989, 2022.

AI Use Disclosure

Large language models were used to assist with language polishing and to draft portions of the manuscript. The authors used these tools to improve readability, revise phrasing, and prepare preliminary text for selected sections. All technical claims, method descriptions, mathematical statements, experimental settings, reported results, and final wording were checked, edited, and approved by the authors, who take full responsibility for the content of the paper.

Appendix A. Multi-prompt frame board

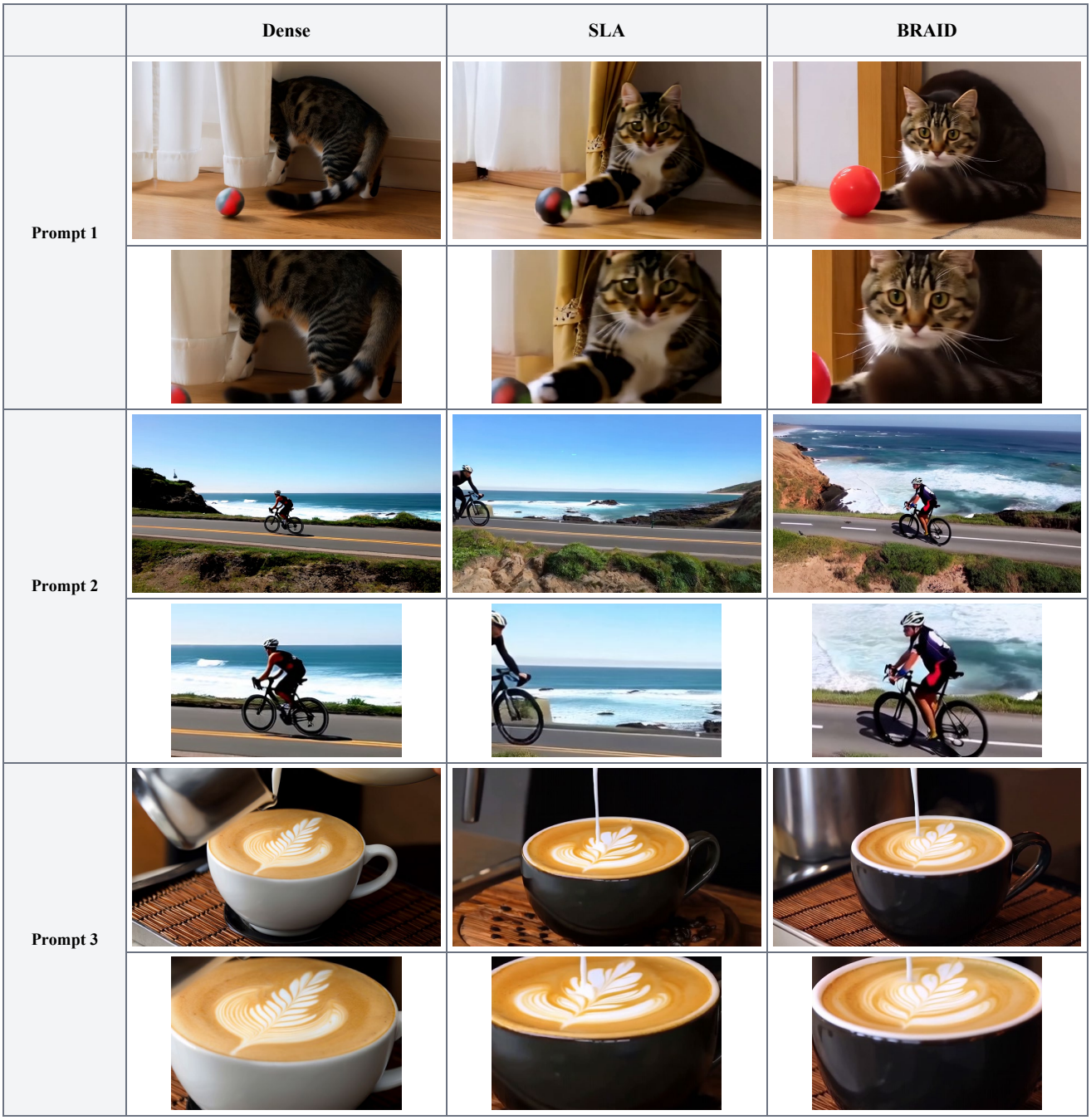

Figure 6. Matched run material: three prompts rendered under one Dense, SLA, and BRAID condition, the detail crop of each prompt shown beneath its key frame. Condition: seed 1, 50 sampler steps at CFG 5.0, nominal block budget 0.15 with control points (0.19, 0.1, 0.2) of trapezoidal mean 0.1475 on the BRAID arm. As with every visual panel in this manuscript, the frames are shown from a single seed for legibility; the scored tables of Section 4.3 average over the 5-seed evaluation set. Prompt-following and appearance scores over the full evaluation set are listed in Table 2.

The generating prompts of the three board rows, quoted in full.

Prompt 1: "A cat crouches low behind a curtain, wiggles its tail, then pounces out toward a rolling ball."

Prompt 2: "A cyclist pedals steadily along a coastal road, leaning into a turn as waves crash below the cliff."

Prompt 3: "A barista pours steamed milk into a cup of espresso, slowly drawing a leaf pattern on the surface."