

Video DeltaNet: A Video-Native Hybrid Attention for Livestream Video Generation

Haocheng Xi^{1,*}, Yiming Xie², Hexu Zhao², Yiwen Zhang², Michael Liu^{1,2}, Thomas Creavin²
Kurt Keutzer¹, Xiuyu Li², Zhaoyang Lv², Chenfeng Xu³, Haiwen Feng^{1,2}

¹University of California, Berkeley ²Impossible, Inc. ³University of Texas at Austin

Video diffusion models repeatedly process long spatiotemporal token sequences during denoising, making attention a major computational bottleneck. Linear attention offers an appealing alternative and has been widely adopted in recent large language models, but directly applying it to video models often fails to preserve the fine-grained interactions required for high-quality generation. We present **Video DeltaNet (VDN)**, which combines local Softmax attention with bidirectional linear memory for long-range video context. Its linear branch introduces **Video Delta Attention (VDA)**, which updates memory once per frame by jointly incorporating its spatial tokens. Separate output projections and learnable gates calibrate the two branches, while a staged teacher-alignment recipe progressively introduces the new pathway into pretrained models. We instantiate VDN on MiniMax H3, applying the hybrid to video-to-video interactions while retaining Softmax for interactions involving text or audio. With eight-step distillation and an optimized SGLang serving stack, VDN-H3 completes DiT denoising for a 14.3-second, 768p video in 6.70 seconds on eight NVIDIA B200 GPUs, corresponding to a $14.5\times$ speedup over the 50-step dense H3 baseline on the same GPU count.

GitHub: <https://github.com/OpenVDN/vdn-minimax-h3>

Weights: <https://huggingface.co/OpenVDN/vdn-minimax-h3>

Blog: <https://openvidn.github.io/>

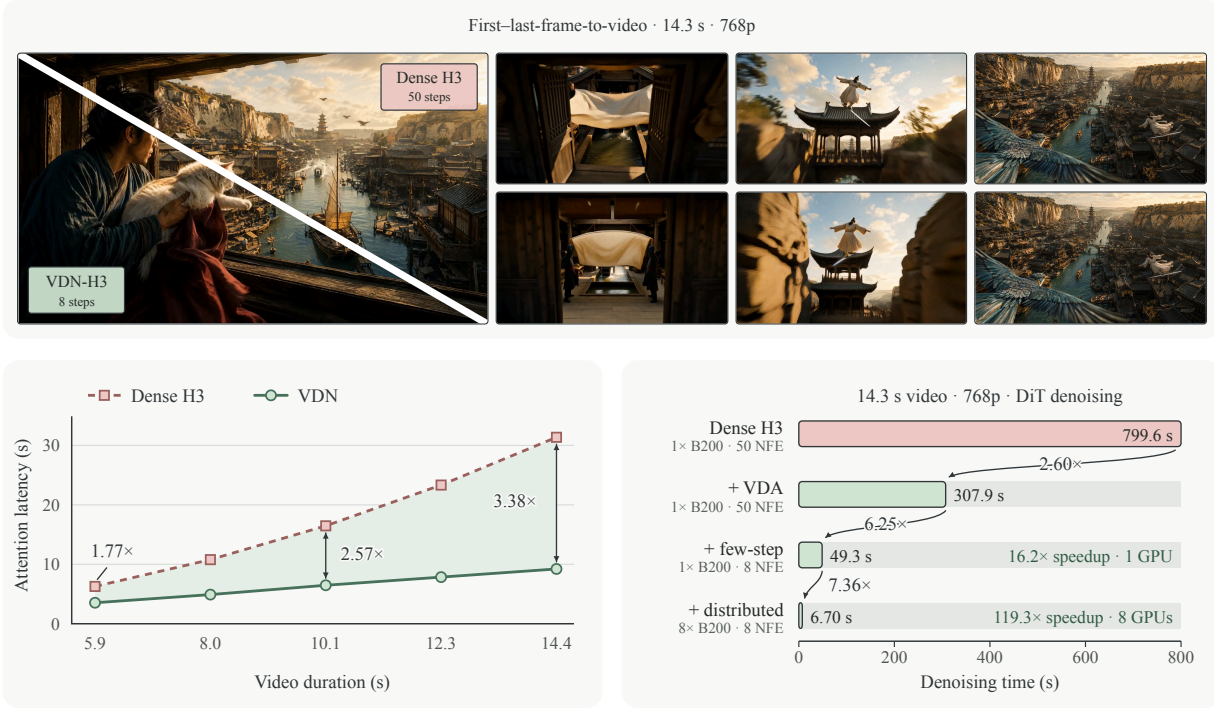


Figure 1 Quality and efficiency of VDN-H3. Top: matched video frames. Bottom: attention scaling and cumulative denoising acceleration.

*Part of the work done during an internship at Impossible, Inc.

1 INTRODUCTION

Attention is the dominant computational cost in long-sequence video generation. In the profiled MiniMax H3 workload, Softmax attention accounts for more than 85% of denoiser runtime. Dense Softmax gives every query direct access to the complete key sequence, but the resulting pairwise computation grows quadratically with sequence length. Frontier language models increasingly use recurrent linear attention to avoid this scaling bottleneck. Applying the same idea to video is appealing: distant context can be compressed into a fixed-size state, making its cost linear in the number of tokens. A direct replacement, however, meaningfully degrades generation quality because the compressed state cannot preserve all of the fine-grained interactions available to Softmax.

Closing this quality gap requires addressing three mismatches. First, a fixed-size state must preserve global properties such as subject identity, scene layout, appearance, and long-range motion even though its capacity does not grow with sequence length, whereas Softmax retains an expanding set of keys and values. Second, delta-rule linear attention typically updates its recurrent state one token at a time, mirroring autoregressive language-model decoding. Video diffusion instead processes all spatial tokens in a frame together; imposing an arbitrary patch order is unnatural, while treating correlated writes independently can make them interfere. Third, adding a randomly initialized linear pathway to a Softmax-pretrained model changes both its information flow and residual-stream activation statistics. Without careful adaptation, it can disrupt capabilities learned during pretraining before the new branch becomes useful.

We introduce **Video DeltaNet (VDN)**, a hybrid attention architecture designed around these challenges. VDN retains Softmax for local video interactions and global boundary anchors, while bidirectional linear memory represents distant video context. The anchors keep the beginning and end of the clip directly accessible as global references. The linear branch uses **Video Delta Attention (VDA)**, a video-native delta rule that updates memory once per frame by jointly incorporating its spatial tokens and key correlations. RMS normalization stabilizes the scale of the linear readout, while separate gates and output projections calibrate each branch before their outputs are combined. A staged adaptation recipe first aligns the new linear pathway with the pretrained teacher, then uses low-rank updates to co-adapt the hybrid layer while preserving the pretrained backbone.

We instantiate the approach on MiniMax H3 to obtain **VDN-H3**. In collaboration with the SGLang team, we optimize its inference path. With eight-step distillation, VDN-H3 completes DiT denoising for a 14.3-second, 768p video in 6.70 seconds on eight NVIDIA B200 GPUs. This corresponds to a $14.5\times$ reduction relative to the 50-step dense H3 baseline on the same GPU count. On a fixed third-party benchmark, eight-step VDN-H3 matches or slightly exceeds 50-step Dense H3 across video-quality metrics, preserves comparable motion magnitude and first-last-frame conditioning fidelity, and maintains a clear margin over four-step FastH3.

Our main contributions are:

1. **A hybrid video attention architecture** that combines local Softmax and global boundary anchors with bidirectional linear memory, together with branch-specific normalization, gates, and output projections.
2. **A frame-wise delta operator** that jointly incorporates a frame’s spatial tokens, with an analysis of the inherited-state transition and an efficient batched implementation.
3. **A complete adaptation recipe for pretrained video models** that combines staged teacher alignment, low-rank refinement, few-step distillation, and optimized inference without training a new foundation model from scratch.

2 VIDEO DELTANET: HYBRID ATTENTION ARCHITECTURE

Video DeltaNet divides video-to-video attention by temporal role. Nearby frames retain explicit token-to-token Softmax attention, while distant video context is summarized by linear attention. Below, we introduce the two branches and explain how their outputs are combined.

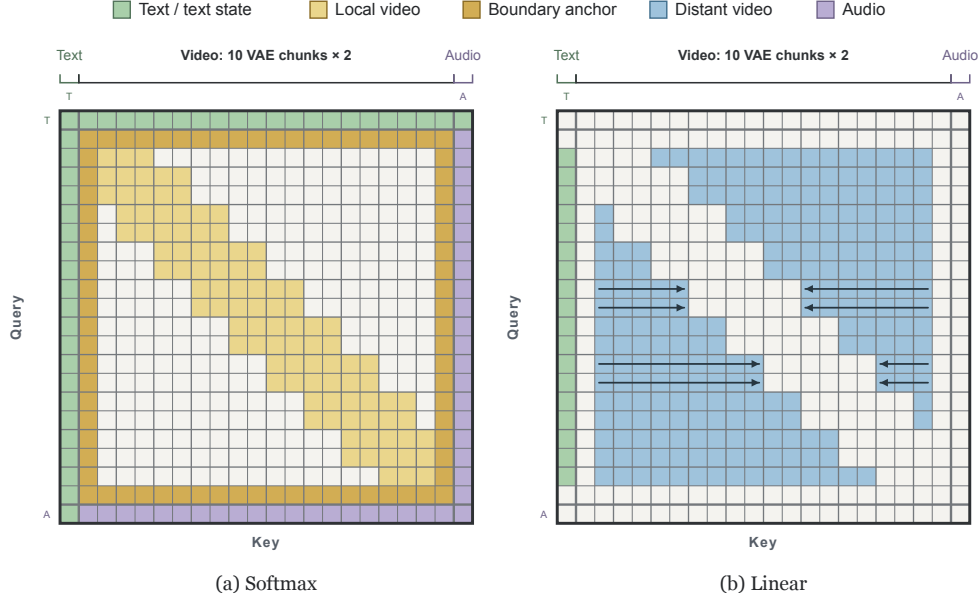


Figure 2: Softmax and Linear attention allocation. The schematic uses one text token, twenty video tokens in ten two-frame chunks, and one audio token.

2.1 SLIDING-WINDOW SOFTMAX ATTENTION

Nearby frames contain local correspondences that determine texture, object boundaries, and short-term motion. These interactions benefit from direct token-to-token matching across neighboring frames. VDN therefore retains exact Softmax attention within a bidirectional temporal window to preserve quality, while assigning distant interactions to linear attention.

The window follows the video tokenizer. H3’s VAE decodes five consecutive latent frames as one temporal chunk, so each query chunk attends to itself and its immediately preceding and following chunks. This prevents the Softmax boundary from cutting through the tokenizer’s natural temporal unit, producing a 15-frame window except at sequence boundaries.

VDN also adds **two boundary anchors with four-way connectivity**. Every video frame attends to all tokens in the first and last latent frames, and the first and last frames attend to the complete sequence. This pattern is particularly natural for full-clip video diffusion: the two boundary anchors provide explicit global references from opposite ends of the clip, while only two frame rows and columns receive dense connectivity. The same anchors are valuable for image-to-video and first-last-frame-to-video generation, where the provided visual conditions directly constrain the generated sequence. Anchor entries already inside a local window are included only once.

2.2 BIDIRECTIONAL LINEAR ATTENTION

Bidirectional linear attention handles the remaining video context through two temporal scans. For a query frame t , the forward state $S_t^{\rightarrow}$ summarizes frames before its Softmax window, and the reverse state $S_t^{\leftarrow}$ summarizes frames after the window. Boundary anchors are excluded from both states because they are already available through Softmax. The two temporal regions are disjoint, so their query readouts can be added without counting any video frame twice.

Each scan first builds frame states along its own direction. At readout time, VDN gathers the state immediately outside the corresponding window boundary, applies the accumulated channel-wise decay across the skipped local span, and evaluates the query against the resulting memory. The distant-context output is the sum of the two readouts:

$$o_t^L = S_t^{\rightarrow} q_t + S_t^{\leftarrow} q_t. \quad (1a)$$

The linear memory is also text-aware. Before scanning the video sequence, VDN summarizes all text tokens into a state S_T and initializes each directional scan with $S_T/2$. Summing the two readouts therefore counts the prompt exactly once, providing the linear branch with global text conditioning while text remains directly visible to the Softmax branch:

$$S_0^{\rightarrow} = S_0^{\leftarrow} = \frac{1}{2} S_T, \quad (S_0^{\rightarrow} + S_0^{\leftarrow}) q = S_T q. \quad (1b)$$

2.3 COMBINING OUTPUTS FROM TWO BRANCHES

Both branches receive their query, key, and value inputs from the pretrained QKV projections. The Softmax branch keeps H3’s QK normalization and rotary position processing. Following Gated DeltaNet (Yang et al., 2025b) and Kimi Delta Attention (Kimi Team, 2025), the linear branch then applies its own feature map: a separable short convolution to K and V, followed by SiLU, with L2 normalization on Q and K. The released K/V convolution consists of a depthwise 5×5 spatial filter and a five-tap temporal filter. No rotary embedding is added in the linear branch.

The two branches have different output scales and therefore require calibration. Restricting Softmax to local windows and boundary anchors concentrates its probability mass over fewer keys, so VDN applies a content-dependent sigmoid gate to the Softmax readout.

The linear readout is RMS-normalized and passed through its own sigmoid output gate. Following Gated DeltaNet and Kimi Delta Attention, separate decay and write gates control memory retention and frame updates inside the recurrence.

Each branch also has its own output projection. Let the gated branch outputs be

$$\tilde{O}^S = G^S \odot O^S, \quad \tilde{O}^L = G^L \odot \text{RMSNorm}(O^L). \quad (1c)$$

The two branches are then projected independently and added:

$$Y = \tilde{O}^S W_O^S + \tilde{O}^L W_O^L. \quad (1d)$$

Separate output projections allow the branches to contribute in different residual-stream directions. The combined output is then fed into the pretrained residual stream, while the original feed-forward sublayer is left unchanged.

3 VIDEO DELTA ATTENTION: FRAME-WISE DELTA RULE

Video Delta Attention (VDA) extends the delta rule from individual tokens to entire video frames. It jointly updates the recurrent state from all key-value pairs in a frame, allowing interactions among frame tokens to shape the write rather than accumulating independent corrections. We first review the standard token-wise rule and then derive the frame-wise update.

3.1 PRELIMINARIES OF LINEAR ATTENTION

Delta-rule memory reads a value associated with a key and writes a correction proportional to the prediction residual. Gated DeltaNet combines this mechanism with memory decay (Yang et al., 2025b), while Kimi Delta Attention introduces finer-grained decay (Kimi Team, 2025). At step t , for a state $S_{t-1} \in \mathbb{R}^{d_v \times d_k}$, a key $k_t \in \mathbb{R}^{d_k}$, and a value $v_t \in \mathbb{R}^{d_v}$, the familiar one-token update is

$$\begin{aligned} \bar{S}_t &= S_{t-1} \text{Diag}(\alpha_t), \\ S_t &= \bar{S}_t + \beta_t (v_t - \bar{S}_t k_t) k_t^\top. \end{aligned} \quad (2)$$

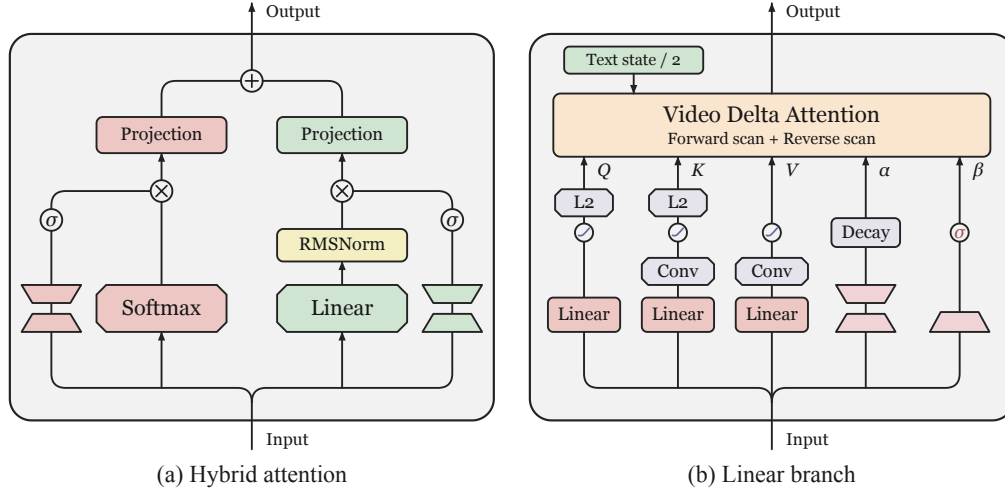


Figure 3: Video DeltaNet architecture. Softmax and Linear outputs are independently gated and projected; the Linear branch uses bidirectional Video Delta Attention.

The decay gate α_t controls how much inherited memory remains, while β_t controls the strength of the erase-and-write correction. This token-wise recurrence is well matched to autoregressive decoding, where one new token arrives at each step.

Video diffusion presents a different unit of computation: all spatial tokens of a latent frame are available together. Sequentially imposing a patch order is unnecessary, so a natural first adaptation is to compute their delta corrections in parallel from the same decayed state. SANA-WM follows this batched construction and adds frame-size key scaling to stabilize the resulting additive transition (Zhu et al., 2026).

Let a video contain F latent frames with $U = H_\ell W_\ell$ tokens each. For token u of frame t , denote its key, value, and write gate by $k_{t,u} \in \mathbb{R}^{d_k}$, $v_{t,u} \in \mathbb{R}^{d_v}$, and $\beta_{t,u} \geq 0$. A frozen-state update adds all token corrections evaluated at $\bar{S}_t = S_{t-1} \text{Diag}(\alpha_t)$:

$$S_t^{\text{batch}} = \bar{S}_t + \sum_{u=1}^U \beta_{t,u} (v_{t,u} - \bar{S}_t k_{t,u}) k_{t,u}^\top. \quad (3a)$$

Stack the keys and values as $K_t \in \mathbb{R}^{U \times d_k}$ and $V_t \in \mathbb{R}^{U \times d_v}$, and write $\beta_t = (\beta_{t,1}, \dots, \beta_{t,U})^\top$. The two frame statistics are

$$\begin{aligned} A_t &= \sum_{u=1}^U \beta_{t,u} k_{t,u} k_{t,u}^\top = K_t^\top \text{Diag}(\beta_t) K_t, \\ B_t &= \sum_{u=1}^U \beta_{t,u} v_{t,u} k_{t,u}^\top = V_t^\top \text{Diag}(\beta_t) K_t. \end{aligned} \quad (3b)$$

Here A_t summarizes key correlations and B_t value-key writes, giving

$$S_t^{\text{batch}} = \bar{S}_t (I - A_t) + B_t. \quad (3c)$$

Because every residual in Equation (3a) uses the same $\bar{S}_t$, overlapping keys can produce conflicting writes without accounting for one another (Appendix A.2).

3.2 VIDEO DELTA ATTENTION: A FRAME-WISE UPDATE

This independence becomes problematic when several patches address similar key directions. Their corrections can reinforce or conflict even though they are meant to describe one frame. VDA instead lets all spatial tokens determine a single new state together, so overlapping directions are resolved inside the update while previously accumulated memory remains a reference.

The classical one-token delta rule can be viewed as one gradient step on its prediction error. Rather than taking one such step independently for every patch, VDA defines the frame-level state as the solution to a joint objective:

$$S_t = \arg \min_S \frac{1}{2} \|S - \bar{S}_t\|_F^2 + \frac{1}{2} \sum_{u=1}^U \beta_{t,u} \|Sk_{t,u} - v_{t,u}\|_2^2. \quad (4)$$

The first term keeps the new memory close to the decayed state. The second asks that same state to fit all key-value associations in the frame simultaneously. Differentiating once gives a compact normal equation and closed-form update:

$$S_t(I + A_t) = \bar{S}_t + B_t, \quad S_t = (\bar{S}_t + B_t)(I + A_t)^{-1}. \quad (5)$$

In contrast to Equation (3c), each residual in the joint solution is evaluated at the shared updated state S_t . The inverse couples the writes: when two patch keys overlap, their inner product affects both effective corrections. We compute this inverse in the $d_k \times d_k$ key-channel space; Appendix A.2 illustrates the effect of key correlations.

3.3 STABILITY AND CORRELATION AWARENESS OF VIDEO DELTA ATTENTION

VDA has a stable inherited-state transition: the contribution carried from earlier frames cannot be amplified by a frame update when the prepared features and gates are fixed.

Proposition 1 — Non-expansive inherited-state transition. For fixed prepared features and gates, with $\beta_{t,u} \geq 0$ and $0 \leq \alpha_{t,j} \leq 1$, the transition $M_t = \text{Diag}(\alpha_t)(I + A_t)^{-1}$ satisfies $\|M_t\|_2 \leq 1$. Thus changing the entering state by ΔS changes its carried contribution by at most $\|\Delta S\|_F$.

Without frame-size scaling, additive frame-wise writes can amplify inherited state; SANA-WM therefore scales its keys by $1/\sqrt{U}$ (Zhu et al., 2026). VDA obtains non-expansiveness directly from $(I + A_t)^{-1}$. Appendix A.1 proves the proposition, and Appendix A.2 compares the two rules under different key correlations.

4 END-TO-END TRAINING PIPELINE

We first describe the three-stage adaptation that integrates the Linear branch into the pretrained Softmax backbone, followed by few-step distillation from 50 to eight denoising steps.

4.1 STAGED ARCHITECTURE ADAPTATION

Architecture adaptation proceeds in three stages. Figure 4 shows the trainable components in each stage; the pretrained base weights remain frozen throughout.

A1 - Per-layer alignment. Each Linear branch is initialized independently from frozen pretrained activations for 200 steps. This local objective avoids sending the initialization signal through a deep stack of simultaneously changing hybrid blocks. The backbone is frozen, the Softmax gate is fixed at its 0.99 initialization, and gradients are clipped independently for each layer at 0.1.

A2 - End-to-end alignment. The calibrated branches are then installed together and optimized end to end for 500 steps. This stage corrects composition errors that are invisible when blocks are trained in isolation. The pretrained weights and Softmax gates remain frozen, and gradient clipping is applied globally at 1.0.

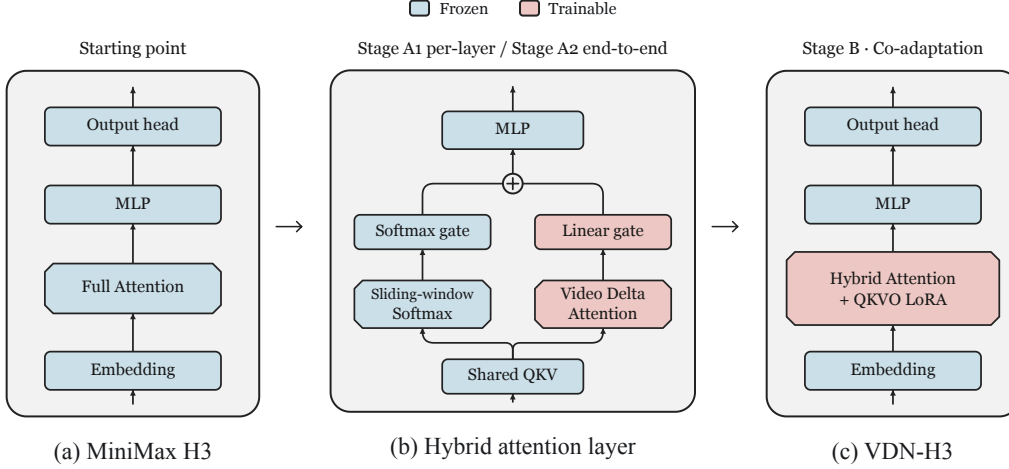


Figure 4: Progressive adaptation of the pretrained H3 backbone. (a) The dense model is the frozen starting point. (b) Stage A1 aligns each Linear branch independently, then Stage A2 trains the assembled hybrid layers end to end while the pretrained weights and Softmax gates remain frozen. (c) Stage B jointly trains the Linear pathway, Softmax gates, and QKVO LoRA adapters.

Stage B - LoRA co-adaptation. We add LoRA adapters to the Q, K, V, and output projections and train them jointly with the Linear pathway and Softmax gates for 2,000 steps.

4.2 FEW-STEP DISTILLATION

After architecture adaptation, we distill the 50-step VDN-H3 into an eight-step sampler using a DMD2-style objective without the GAN term (Yin et al., 2024b). The student is initialized from the community MiniMax-H3-Turbo-LoRA (larryvrh, 2026) and trained against VDN-H3’s own 50-step sampler, isolating step reduction from architecture conversion. Generator, real-score, and fake-score roles share one FSDP backbone through separate adapters, with three fake-score updates per generator update. The released checkpoint is trained for 250 generator steps.

5 EFFICIENT INFERENCE

Fused VDA kernels. We organize VDA’s data preparation and readout into four fused Triton kernels. *VDA-Prep* combines temporal convolution, SiLU, L2 normalization, and the frame-major layout conversion for K and V. *VDA-Stats* constructs the per-frame statistics A_t and B_t in one pass, sharing input reads and reduction work. *VDA-Gather* collects the two directional states at local-window boundaries and applies the decay bridge. *VDA-Epilogue* combines RMS normalization, output gating, and the final layout conversion. QK normalization and rotary embedding are fused separately on the Softmax path.

Chunk-wise scans. VDA defines an affine transition per frame, but attention reads memory only at VAE-chunk boundaries. We therefore compose each chunk into $S_{\text{out}} = S_{\text{in}}M_{\text{chunk}} + J_{\text{chunk}}$ and scan the shorter chunk sequence in both directions with a single kernel launch. This preserves the required boundary states while reducing scan depth and launch overhead by roughly the chunk size. The prompt state is included as a leading virtual frame.

Small-matrix inverse. Each frame requires computing $(I + A_t)^{-1}$. A batched Cholesky pipeline incurs several kernel launches and intermediate memory reads and writes, so we use one CUDA kernel that performs blocked Gauss–Jordan elimination in registers and directly emits the transition and injection terms. The inverse and recurrent state updates remain in FP32.

Other optimizations. The released VDN-H3 model is served through SGLang. Window Softmax packs queries by visible-key pattern to use FlashAttention’s varlen API without a global mask. Following Ulysses (Jacobs et al., 2023), VDA is sharded by attention head and executed on a side stream

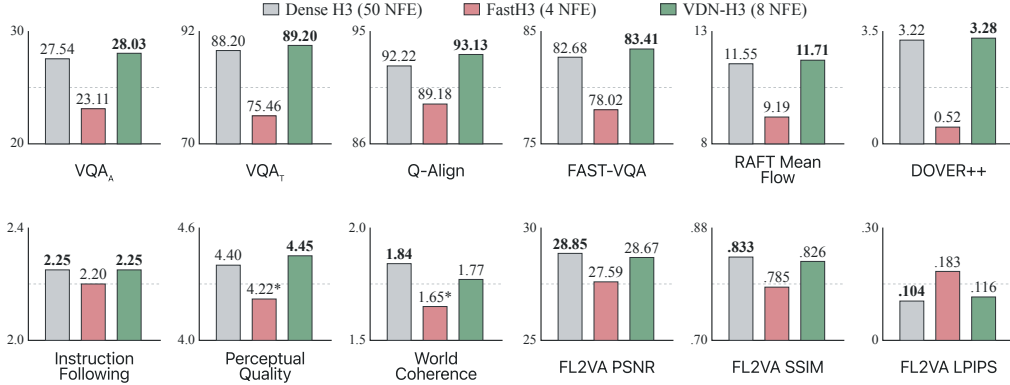


Figure 5: Video quality, motion, FIRM-Video dimensions, and FL2VA endpoint fidelity across Dense H3, FastH3, and VDN-H3. FL2VA metrics average the first and last conditioning frames; * denotes a reported significant difference from Dense H3.

alongside window Softmax. MXFP8 accelerates the wide QKV, output, and feed-forward GEMMs, while recurrent states and small-matrix inverses remain in FP32. AdaLN modulation parameters are precomputed before the block loop to avoid repeating their projection inside each transformer block.

6 EXPERIMENTS

6.1 SETTINGS

Data. We curated a training set of 10,015 video clips at 1344×768 resolution, each with 345 frames at 24 fps (14.375 seconds). Each sample is pre-encoded and cached as video latents of shape (24, 102, 48, 84), stereo audio latents of shape (2, 32, 575), and Qwen3-VL text embeddings of shape (L , 5120) with token-type tags (Bai et al., 2025). This removes the VAEs and text encoder from the training loop. For evaluation, we use 103 prompts from a fixed third-party set. All models render the same prompts at the same resolution and duration, without model-specific prompt selection.

Baselines. Dense H3 with 50 neural function evaluations (NFEs) is the full-attention baseline, while FastH3 with four NFEs provides a fast-model baseline (FastVideo Team, 2026). Quality and qualitative comparisons use the final eight-step VDN-H3. We report 50-step VDN-H3 only in the efficiency study, where it isolates the speedup from hybrid attention before step distillation.

Metrics. We report EvalCrafter VQA_A and VQA_T (Liu et al., 2024), Q-Align (Wu et al., 2024), FAST-VQA (Wu et al., 2022), and DOVER++ overall (Wu et al., 2023). Higher is better for all five; Q-Align, FAST-VQA, and DOVER++ use a $\times 100$ scale. We additionally report RAFT mean flow magnitude in pixels as a motion diagnostic (Teed & Deng, 2020). FIRM-Video evaluates Instruction Following, Perceptual Quality (PQ), and World Coherence (WC) on a 1–5 scale (Zhang et al., 2026d). For first–last–frame-to-video (FL2VA), we measure PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) at both conditioning frames and report their mean.

Environment. We use PyTorch 2.13 and CUDA 12.9 on NVIDIA H200 and B200 clusters. Training uses FSDP2/HSDP, activation checkpointing, and pinned-memory activation offload. Parameters are gathered in bf16 and gradients reduced in FP32, except that the decay- α modules remain FP32. Additional optimization and training details are provided in Appendix A.4.

6.2 QUALITY RESULTS

Figure 5 summarizes overall video quality, motion, and conditional endpoint fidelity. Across the five no-reference quality metrics, eight-step VDN-H3 matches or exceeds 50-step Dense H3, with differences ranging from +0.06 to +1.00; FastH3 is 2.70–12.74 points lower than Dense H3. This agreement holds across aesthetic, technical, and learned perceptual metrics. VDN-H3 also preserves a similar RAFT motion magnitude (11.71 versus 11.55 pixels), while FastH3 falls to 9.19

Table 2: B200 efficiency. Gain is measured against the preceding row; speedup is cumulative against single-GPU Dense H3. The video-duration sweep uses one GPU and excludes distributed inference.

Configuration	#GPU	NFE	s/NFE	End-to-end latency	Gain	Speedup	Latent frames	42	72	87	102
Dense H3	1	50	15.99 s	799.6 s	–	1.0×	Attention density	42.08%	26.82%	22.85%	19.98%
VDN-H3	1	50	6.16 s	307.9 s	2.6×	2.6×	Attention speedup	1.7×	2.3×	2.7×	3.0×
+ few-step	1	8	6.16 s	49.3 s	6.3×	16.2×	End-to-end latency	19.0 s	34.1 s	41.6 s	49.3 s
+ distributed	8	8	0.85 s	6.70 s	7.4×	119.3×	End-to-end speedup	10.3×	13.2×	14.7×	16.2×

Table 3: H200 efficiency. Gain is measured against the preceding row; speedup is cumulative against single-GPU Dense H3. The video-duration sweep uses one GPU and excludes distributed inference.

Configuration	#GPU	NFE	s/NFE	End-to-end latency	Gain	Speedup	Latent frames	42	72	87	102
Dense H3	1	50	35.35 s	1767.3 s	–	1.0×	Attention density	42.08%	26.82%	22.85%	19.98%
VDN-H3	1	50	11.16 s	557.9 s	3.2×	3.2×	Attention speedup	2.0×	3.0×	3.5×	4.0×
+ few-step	1	8	11.16 s	89.3 s	6.3×	19.8×	End-to-end latency	34.9 s	62.2 s	76.3 s	89.3 s
+ distributed	8	8	1.56 s	12.5 s	7.1×	141.6×	End-to-end speedup	11.4×	15.6×	17.7×	19.8×

pixels. FIRM-Video shows the same pattern: VDN-H3 matches Dense H3 in Instruction Following (2.25), is slightly higher in Perceptual Quality (4.45 versus 4.40), and remains comparable in World Coherence (1.77 versus 1.84), whereas FastH3 is lower on all three dimensions.

For FL2VA, VDN-H3 remains close to Dense H3, with gaps of 0.18 dB in PSNR (28.67 versus 28.85), 0.007 in SSIM (0.826 versus 0.833), and 0.011 in LPIPS (0.1156 versus 0.1044; lower is better). FastH3 shows a larger loss of endpoint fidelity: relative to Dense H3, its PSNR and SSIM decrease by 1.26 dB and 0.048, while LPIPS increases by 0.079.

6.3 EFFICIENCY RESULTS

Backbone acceleration. We first compare one complete transformer evaluation at the same sequence length and NFE count. For the 14.3-second, 768p workload, the system-optimized VDN-H3 backbone reduces latency from 35.35 to 11.16 seconds on one H200 (3.2×), and from 16.0 to 6.2 seconds on one B200 (2.6×).

Scaling with video length. Dense Softmax scores every video-token pair, so its video–video workload grows quadratically with the number of latent frames. VDN keeps only a fixed-width local window and the first- and last-frame anchors in Softmax, while VDA carries the remaining long-range context with linear scaling. As the sequence grows from 42 to 102 latent frames, the measured Softmax attention density falls from 42.1% to 20.0%. Over the same range, whole-backbone speedup increases from 1.8× to 3.2× on H200 and from 1.7× to 2.6× on B200.

Sampling and distributed inference. Starting from the optimized backbone, eight-step distillation reduces one-B200 denoising from 307.9 to 49.3 seconds, and eight-GPU head-sharded inference brings the final latency to 6.7 seconds on B200 and 12.5 seconds on H200. Tables 2 and 3 report the complete deployment path and duration scaling on both accelerators.

6.4 ABLATION STUDIES

We profile VDN’s core backbone operators at 102 latent frames. Figure 6 compares their direct and optimized implementations on H200 and B200, separating operator-level gains from few-step distillation and distributed inference.

Fused VDA kernels. *VDA-Prep* reduces latency from 18.0 to 1.6 ms on H200 and from 17.1 to 3.4 ms on B200. *VDA-Stats* provides 2.1× and 2.9× speedups, *VDA-Gather* provides 7.2× and 7.5×, and *VDA-Epilogue* provides 7.3× and 9.1× on H200 and B200, respectively.

Chunk-wise scans. Composing frame transitions adds overhead with all 56 heads on one GPU, but becomes effective after head sharding. With seven heads per rank in 8-GPU inference, scan latency falls from 4.6 to 1.1 ms on H200 and from 3.0 to 0.8 ms on B200.

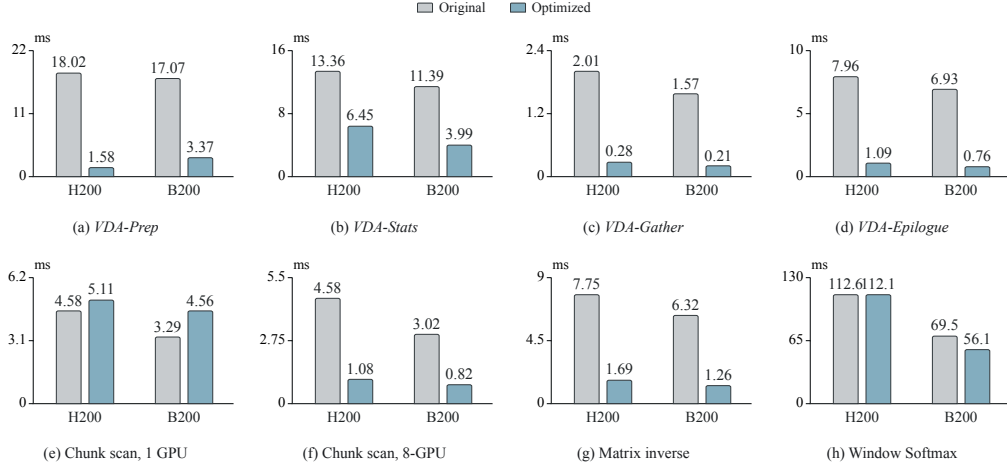


Figure 6: Kernel-level optimization ablations on H200 and B200.

Small-matrix inverse. Replacing the multi-kernel Cholesky path with the fused inverse reduces latency from 7.8 to 1.7 ms on H200 and from 6.3 to 1.3 ms on B200, a 4.6–5.0 $\times$ speedup.

Other optimizations. Figure 6 additionally isolates Window Softmax. It improves from 69.5 to 56.1 ms on B200, while H200 remains nearly unchanged at 112.6 versus 112.1 ms.

6.5 GALLERY

In Figure 7, we present diverse examples comparing 50-step Dense H3 and eight-step VDN-H3 across visual styles, scene transitions, and motion patterns. The results demonstrate that VDN-H3 achieves visual quality on par with Dense H3 despite using substantially fewer denoising steps.

7 RELATED WORK

Efficient video attention algorithms. Recent video diffusion transformers process increasingly long spatiotemporal sequences (HaCohen et al., 2024; Jin et al., 2024; Kong et al., 2024; Polyak et al., 2024; Wan et al., 2025; Yang et al., 2025c; MiniMax, 2026). Attention acceleration spans optimized exact kernels (Dao et al., 2022; Dao, 2023; Shah et al., 2024) and low-precision kernels, including SageAttention (Zhang et al., 2025b), SageAttention2 (Zhang et al., 2024), and SageAttention3 (Zhang et al., 2025a). General-purpose sparse methods include SpargeAttn (Zhang et al., 2025d) and SpargeAttention2 (Zhang et al., 2026b); video-specific methods include Sparse VideoGen (Xi et al., 2025) and Sparse VideoGen2 (Yang et al., 2025a). SLA and SLA2 combine sparse and linear computation (Zhang et al., 2025c, 2026a). Quantized diffusion methods reduce the cost of DiT execution (Zhao et al., 2024; Li et al., 2025, 2026; Xue et al., 2026), while Quant VideoGen targets the recurrent cache of autoregressive video models (Xi et al., 2026).

Linear attention and recurrent memory. Linear Transformers and Performers introduced influential linear-complexity alternatives to dense attention (Katharopoulos et al., 2020; Choromanski et al., 2021), while Mamba and Mamba-2 connected selective state-space models with efficient sequence processing (Gu & Dao, 2023; Dao & Gu, 2024). Gated Linear Attention (Yang et al., 2024b), DeltaNet (Yang et al., 2024a), Gated Delta Networks (Yang et al., 2025b), Kimi Linear (Kimi Team, 2025), and Gated DeltaNet-2 (Hatamizadeh et al., 2026) support data-dependent memory updates. In vision, VideoMamba applies state-space modeling to long-video understanding, and M4V develops a multimodal Mamba backbone for text-to-video generation (Li et al., 2024b; Huang et al., 2026). LoGeR similarly combines sliding-window attention with parametric long-term memory to preserve local geometry and global consistency over long video sequences (Zhang et al., 2026c). Linear and hybrid diffusion backbones include SANA (Xie et al., 2025), SANA-Video (Chen et al., 2025), SANA-Video 2.0 (Chen et al., 2026), and SANA-WM (Zhu et al., 2026). We also acknowl-



Figure 7: Qualitative gallery. Six synchronized frames from five prompts, with 50-step Dense H3 above eight-step VDN-H3 in each pair.

edge Reflections on Video DeltaNet as concurrent analysis of frame-level delta updates (Zhu, 2026).

Few-step distillation and distributed inference. Progressive distillation, consistency models, latent consistency models, and adversarial distillation reduce the number of denoising evaluations (Salimans & Ho, 2022; Song et al., 2023; Luo et al., 2023; Sauer et al., 2023). Distribution Matching Distillation (Yin et al., 2024a) and DMD2 (Yin et al., 2024b) provide closely related objectives; AnimateLCM and T2V-Turbo extend few-step training to video (Wang et al., 2024; Li et al., 2024a). Multi-GPU inference is complementary: Ulysses and USP shard long attention sequences (Jacobs et al., 2023; Fang & Zhao, 2024), DistriFusion and PipeFusion exploit patch and pipeline parallelism across diffusion steps (Li et al., 2024c; Fang et al., 2024a), and xDiT composes multiple

parallel strategies in one inference engine (Fang et al., 2024b). StreamDiffusionV2 further targets low-latency distributed video streaming (Feng et al., 2026).

8 CONCLUSION

Video DeltaNet accelerates video generation while largely preserving Dense H3 quality. On the 14.3-second, 768p workload, its optimized backbone is $2.6\times$ faster on one B200 and $3.2\times$ faster on one H200. With eight-step distillation and eight-GPU inference, DiT denoising takes 6.70 seconds on B200s and 12.5 seconds on H200s. Eight-step VDN-H3 remains close to 50-step Dense H3 across the reported quality evaluations.

REFERENCES

- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.
- Junsong Chen, Yuyang Zhao, Jincheng Yu, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Daquan Zhou, Huan Ling, et al. SANA-Video: Efficient video generation with block linear diffusion transformer. *arXiv preprint arXiv:2509.24695*, 2025.
- Junsong Chen et al. SANA-Video 2.0: Hybrid linear attention with attention residuals for efficient video generation. *arXiv preprint arXiv:2607.21553*, 2026.
- Krzysztof Choromanski et al. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.
- Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.
- Tri Dao and Albert Gu. Transformers are SSMS: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*, 2024.
- Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.
- Jiarui Fang and Shangchun Zhao. USP: A unified sequence parallelism approach for long context generative ai. *arXiv preprint arXiv:2405.07719*, 2024.
- Jiarui Fang, Jinzhe Pan, Aoyu Li, Xibo Sun, and Jiannan Wang. PipeFusion: Patch-level pipeline parallelism for diffusion transformers inference. *arXiv preprint arXiv:2405.14430*, 2024a.
- Jiarui Fang, Jinzhe Pan, Xibo Sun, Aoyu Li, and Jiannan Wang. xDiT: An inference engine for diffusion transformers with massive parallelism. *arXiv preprint arXiv:2411.01738*, 2024b.
- FastVideo Team. FastH3 Preview v1: Open-weight 4-step sparse-distilled MiniMax-H3. Hugging Face model repository, 2026. URL <https://huggingface.co/FastVideo/FastVideo-FastH3-4-step-Preview-v1-LoRA>.
- Tianrui Feng, Zhi Li, Shuo Yang, Haocheng Xi, Muyang Li, Xiuyu Li, Lvmin Zhang, Keting Yang, Kelly Peng, Song Han, Maneesh Agrawala, Kurt Keutzer, Akio Kodaira, and Chenfeng Xu. StreamDiffusionV2: A streaming system for dynamic and interactive video generation. In *Proceedings of Machine Learning and Systems*, 2026.
- Chongjian Ge, Hanwen Jiang, Tianyu Wang, Jiuxiang Gu, Yiran Xu, Ziwen Chen, Shaoteng Liu, Jing Shi, Yicong Hong, Zefan Cai, Hailin Jin, and Hao Tan. Chimera: Designing and chinchilla-scaling hybrid visual diffusion transformers. *arXiv preprint arXiv:2607.28611*, 2026.
- Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- Yoav HaCohen et al. LTX-Video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.
- Ali Hatamizadeh, Yejin Choi, and Jan Kautz. Gated DeltaNet-2: Decoupling erase and write in linear attention. *arXiv preprint arXiv:2605.22791*, 2026.

- Jiancheng Huang, Gengwei Zhang, Zequn Jie, Siyu Jiao, Yinlong Qian, Ling Chen, Yunchao Wei, and Lin Ma. M4V: Multimodal mamba for efficient text-to-video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.
- Sam Ade Jacobs et al. DeepSpeed Ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.
- Yang Jin et al. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 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*, 2020.
- Kimi Team. Kimi linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692*, 2025.
- Weijie Kong et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.
- larryvrh. MiniMax-H3-Turbo-LoRA. Hugging Face model repository, 2026. URL <https://huggingface.co/larryvrh/MiniMax-H3-Turbo-Lora>.
- Jiachen Li et al. T2V-Turbo: Breaking the quality bottleneck of video consistency model with mixed reward feedback. *arXiv preprint arXiv:2404.08865*, 2024a.
- Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. VideoMamba: State space model for efficient video understanding. *arXiv preprint arXiv:2403.06977*, 2024b.
- Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Ming-Yu Liu, Kai Li, and Song Han. DistriFusion: Distributed parallel inference for high-resolution diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024c.
- Muyang Li et al. SVDQuant: Absorbing outliers by low-rank components for 4-bit diffusion models. In *International Conference on Learning Representations*, 2025.
- Xingyang Li et al. DeltaQuant: 4-bit video diffusion models with spatiotemporal delta smoothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.
- Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tiejong Zeng, Raymond Chan, and Ying Shan. EvalCrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- Simian Luo et al. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.
- MiniMax. MiniMax H3: An Open Model Breaking the Boundaries Between Tasks and Modalities. Model release, 2026. URL <https://www.minimax.io/blog/minimax-h3>.
- Adam Polyak et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.
- Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.
- Axel Sauer et al. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.
- Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. In *Advances in Neural Information Processing Systems*, 2024.
- Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023.
- Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, 2020.
- Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Fei Wu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

- Fu-Yun Wang et al. AnimateLCM: Accelerating the animation of personalized diffusion models and adapters with decoupled consistency learning. *arXiv preprint arXiv:2402.00769*, 2024.
- Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- Haoning Wu, Chaofeng Chen, Jingwen Hou, Liang Liao, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. FAST-VQA: Efficient end-to-end video quality assessment with fragment sampling. In *European Conference on Computer Vision*, 2022.
- Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.
- Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, Qiong Yan, Xiongkuo Min, Guangtao Zhai, and Weisi Lin. Q-Align: Teaching LMMs for visual scoring via discrete text-defined levels. In *International Conference on Machine Learning*, 2024.
- Haocheng Xi, Shuo Yang, Yilong Zhao, Muyang Li, Han Cai, Xingyang Li, Yujun Lin, Zhuoyang Zhang, Jintao Zhang, Xiuyu Li, Zhiying Xu, Jun Wu, Chenfeng Xu, Ion Stoica, Song Han, and Kurt Keutzer. Quant VideoGen: Auto-regressive long video generation via 2-bit KV-cache quantization. *arXiv preprint arXiv:2602.02958*, 2026.
- Haocheng Xi et al. Sparse VideoGen: Accelerating video diffusion transformers with spatial-temporal sparsity. *arXiv preprint arXiv:2502.01776*, 2025.
- Enze Xie et al. SANA: Efficient high-resolution image synthesis with linear diffusion transformers. In *International Conference on Learning Representations*, 2025.
- Bowen Xue, Zihan Min, Xingyang Li, Zhekai Zhang, Haocheng Xi, Lvmin Zhang, Maneesh Agrawala, Jun-Yan Zhu, Song Han, Yujun Lin, and Muyang Li. FourTune: Towards fully 4-bit efficient post-training for diffusion models. *arXiv preprint arXiv:2607.05711*, 2026.
- Shuo Yang et al. Sparse VideoGen2: Accelerate video generation with sparse attention via semantic-aware permutation. *arXiv preprint arXiv:2505.18875*, 2025a.
- Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, 2025b.
- Songlin Yang et al. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems*, 2024a.
- Songlin Yang et al. Gated linear attention transformers with hardware-efficient training. In *International Conference on Machine Learning*, 2024b.
- Zhuoyi Yang et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, 2025c.
- Tianwei Yin et al. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024a.
- Tianwei Yin et al. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024b.
- Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. SageAttention2: Efficient attention with thorough outlier smoothing and per-thread INT4 quantization. *arXiv preprint arXiv:2411.10958*, 2024.
- Jintao Zhang, Jia Wei, Haoxu Wang, Pengle Zhang, Xiaoming Xu, Haofeng Huang, Kai Jiang, Jianfei Chen, and Jun Zhu. SageAttention3: Microscaling FP4 attention for inference and an exploration of 8-bit training. *arXiv preprint arXiv:2505.11594*, 2025a.
- Jintao Zhang et al. SageAttention: Accurate 8-bit attention for plug-and-play inference acceleration. In *International Conference on Learning Representations*, 2025b.

- Jintao Zhang et al. SLA: Beyond sparsity in diffusion transformers via fine-tunable sparse-linear attention. *arXiv preprint arXiv:2509.24006*, 2025c.
- Jintao Zhang et al. SpargeAttn: Accurate sparse attention accelerating any model inference. In *International Conference on Machine Learning*, 2025d.
- Jintao Zhang et al. SLA2: Sparse-linear attention with learnable routing and QAT. *arXiv preprint arXiv:2602.12675*, 2026a.
- Jintao Zhang et al. SpargeAttention2: Trainable sparse attention via hybrid top-k+top-p masking and distillation fine-tuning. *arXiv preprint arXiv:2602.13515*, 2026b.
- Junyi Zhang, Charles Herrmann, Junhwa Hur, Chen Sun, Ming-Hsuan Yang, Forrester Cole, Trevor Darrell, and Deqing Sun. LoGeR: Long-context geometric reconstruction with hybrid memory. *arXiv preprint arXiv:2603.03269*, 2026c.
- Peiyuan Zhang, Xiangyu Zhao, Hongbo Liu, Xiaoxing Hu, Mingxin Liu, Shuran Ma, Yunhang Shen, Jian Hu, Haihan Gao, Haoyu Cao, and Xue Yang. FIRM-Video: Check before you score for reliable text-to-video reward modeling. *arXiv preprint arXiv:2608.21839*, 2026d.
- Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- Tianchen Zhao et al. ViDiT-Q: Efficient and accurate quantization of diffusion transformers for image and video generation. *arXiv preprint arXiv:2406.02540*, 2024.
- Haoyi Zhu. Reflections on Video DeltaNet. Blog post, aug 2026. URL <https://haoyizhu.site/blog/video-delta-rule/>.
- Haoyi Zhu et al. SANA-WM: Efficient minute-scale world modeling with hybrid linear diffusion transformer. *arXiv preprint arXiv:2605.15178*, 2026.

A DERIVATION AND IMPLEMENTATION DETAILS

A.1 PROOF OF PROPOSITION 1

Fix the prepared features and gates of frame t . Substituting $\bar{S}_t = S_{t-1} \text{Diag}(\alpha_t)$ into Equation (5) separates the inherited state from the new write:

$$\begin{aligned} S_t &= S_{t-1}M_t + J_t, \\ M_t &= \text{Diag}(\alpha_t)(I + A_t)^{-1}, \quad J_t = B_t(I + A_t)^{-1}. \end{aligned} \quad (\text{A.1})$$

For two entering states with the same frame inputs, $\Delta S_t = \Delta S_{t-1}M_t$. Since $\beta_{t,u} \geq 0$, the Gram matrix is positive semidefinite: $x^\top A_t x = \sum_u \beta_{t,u} (k_{t,u}^\top x)^2 \geq 0$ for every x . Writing $A_t = Q \text{Diag}(\lambda_i) Q^\top$, with $\lambda_i \geq 0$, shows that $(I + A_t)^{-1}$ has eigenvalues $1/(1 + \lambda_i) \in (0, 1]$. Together with $0 \leq \alpha_{t,j} \leq 1$, this gives

$$\begin{aligned} \|\Delta S_t\|_F &\leq \|\Delta S_{t-1}\|_F \|M_t\|_2, \\ \|M_t\|_2 &\leq \|\text{Diag}(\alpha_t)\|_2 \|(I + A_t)^{-1}\|_2 \leq 1. \end{aligned} \quad (\text{A.2})$$

This proves Proposition 1 without requiring the decay and Gram matrix to commute. The bound concerns inherited state at fixed frame inputs, not new writes or the full network.

A.2 FRAME-SIZE SCALING AND CORRELATION AWARENESS

The difference between independent and joint frame writes can be seen by expanding Equation (3c):

$$S_t^{\text{batch}} = \bar{S}_t + \sum_{u=1}^U \beta_{t,u} (v_{t,u} - \bar{S}_t k_{t,u}) k_{t,u}^\top.$$

All corrections use the same pre-update state, so a patch’s residual ignores the other current-frame patches.

By comparison, the first-order condition for VDA can be rearranged as

$$S_t = \bar{S}_t + \sum_{u=1}^U \beta_{t,u} (v_{t,u} - S_t k_{t,u}) k_{t,u}^\top.$$

Here every residual depends on the shared S_t . Overlapping keys accumulate in A_t , so Equation (5) adjusts their joint update.

With $\alpha_t = 1$, the additive inherited-state factor is $I - A_t$. Its eigenvalues $1 - \lambda_i(A_t)$ leave $[-1, 1]$ if any $\lambda_i(A_t) > 2$. For unit keys and sigmoid write gates, $\lambda_{\max}(A_t) \leq \text{tr}(A_t) = \sum_u \beta_{t,u} \leq U$, and aligned keys can approach this bound. Thus per-token normalization alone does not ensure stability.

SANA-WM scales each unit key by an additional $1/\sqrt{U}$ (Zhu et al., 2026), replacing A_t by A_t/U so that $I - A_t/U$ is non-expansive. VDA instead uses $(I + A_t)^{-1}$ and needs no frame-size scale.

The inverse also responds to the observed key correlations. If all U positions repeat a unit key k with a common gate β , writing $s = \bar{S}_t k$ and $\bar{v} = U^{-1} \sum_u v_u$ gives

$$S_t k = \frac{1}{1 + U\beta} s + \frac{U\beta}{1 + U\beta} \bar{v}. \quad (\text{A.3})$$

Repeated keys thus receive a saturating $U\beta/(1 + U\beta)$ weight, whereas orthogonal keys each receive $\beta/(1 + \beta)$. SANA-WM’s scaled-key readout instead assigns β to repeated-key consensus and β/U to each orthogonal direction. VDA adapts to key geometry rather than applying the same U -based scale.

A.3 BOUNDARY GATHER AND DECAY BRIDGE

After excluding anchor indices 0 and $F - 1$, let P_j be the forward state through interior frame j and R_j the reverse state from the last interior frame down to j . Their virtual initial states are $P_0 = R_{F-1} = S_T/2$. For query t , let $[\ell_t, h_t]$ be its local window clipped to interior frame indices. The state read by its prepared queries is

$$\begin{aligned}\tilde{S}_t &= P_{\ell_t-1} \text{Diag}\left(\prod_{r=\ell_t}^t \alpha_r\right) \\ &\quad + R_{h_t+1} \text{Diag}\left(\prod_{r=t}^{h_t} \alpha_r\right), \\ o_{t,u}^L &= \tilde{S}_t q_{t,u}.\end{aligned}\tag{A.4}$$

The products are channel-wise. Missing distant video on one side selects the corresponding initial text state rather than a fabricated frame state. The bridge applies decay without local frame writes. If a window covers the complete clip, the implementation bypasses the linear pathway and uses the dense attention path. Clips consisting only of anchor frames also receive no linear video output.

Algorithm 1. Bidirectional VDA inside one hybrid attention block

1. Compute shared QKV and retained multimodal Softmax with chunk and anchor masks.
2. Remove anchor video rows from the linear inputs; prepare Q/K/V features and frame gates.
3. Compute A_t, B_t, α_t for all interior frames. Form the text state from the prompt.
4. Perform a batched factorization of $I + A_t$; construct C_t, M_t, J_t for all frames and heads.
5. Scan $S \leftarrow SM_t + J_t$ in both temporal directions, each initialized with $S_T/2$.
6. Gather outside-window states and apply the decay bridge in Equation (A.4).
7. Read each query; apply RMSNorm and the linear gate; zero anchor rows.
8. Gate and project Softmax; add the projected linear output to video rows.

A.4 ARCHITECTURE ADAPTATION HYPERPARAMETERS

We use AdamW with linear warmup followed by cosine decay. Stages A1, A2, and B use a peak learning rate of 10^{-4} and decay to 5×10^{-6} ; Stage D uses 10^{-5} for the generator and 2×10^{-5} for the fake-score model, decaying to 10^{-6} and 2×10^{-6} , respectively. Following Chimera (Ge et al., 2026), parameters are divided into learning-rate groups by module scale: large fan-in matrices use the base schedule, while smaller vector parameters use larger multipliers. In Stage B, the Linear branch uses $0.4\times$ the LoRA learning rate. The table below lists the remaining stage-specific settings.

Table 4: Hyperparameters for architecture adaptation. The table covers Stages A1, A2, and B; few-step distillation is described in Section 4.2.

Setting	A1: per-layer	A2: end-to-end	B: LoRA co-adaptation
Parameter status	Train: one Linear branch Freeze: backbone and Softmax gate	Train: all Linear branches Freeze: backbone and Softmax gates	Train: branches, gates, and QKVO LoRA Freeze: backbone FFN, embeddings, and head
Steps	200	500	2,000
Optimizer / schedule	AdamW; linear warmup $\rightarrow$ cosine	AdamW; linear warmup $\rightarrow$ cosine	AdamW; linear warmup $\rightarrow$ cosine
β_1, β_2	0.9, 0.999	0.9, 0.999	0.9, 0.999
Peak / minimum LR	$10^{-4} / 5 \times 10^{-6}$	$10^{-4} / 5 \times 10^{-6}$	$10^{-4} / 5 \times 10^{-6}$
Warmup steps	50	50	200
Weight decay	0	0	0
Gradient clip	0.1 per layer	1.0 global	1.0 global
LR multipliers	large 1.0 / small 5.0	large 1.0 / small 5.0	LoRA 1.0 / branch 0.4 / small 2.0
LoRA rank / scale	–	–	64 / 64

A.5 RELEASE CONFIGURATION AND REPRODUCIBILITY

The architectural settings used in this report correspond to the released `vdn_solve` branch, chunk size 5 with radius 1, anchors in both row and column directions, K/V short convolution, text-state initialization, and the alpha decay bridge. Architecture metadata is created in A1 and inherited from the checkpoint in later stages. This is preferable to silently changing an attention mask or state rule when resuming a run.

The release package will include exact weight hashes, dataset and evaluation-prompt manifests, random seeds, tokenizer and frame-alignment metadata, GPU topology, software versions, timing protocol, and quality-evaluation outputs. These artifacts connect each reported result to a reproducible model and execution configuration.

A.6 QUALITY ACROSS DENOISING BUDGETS

Figures 8–10 compare the same evaluation suite at 50, eight, and four neural function evaluations (NFEs). Each metric uses the same labeled, truncated vertical scale across the three figures, but different metrics have different scales. An asterisk reproduces the reported paired significance flag relative to 50-step Dense H3; it does not establish significance between two non-dense variants. RAFT mean flow is a motion-magnitude diagnostic rather than a quality score, and lower FL2VA LPIPS is better.

Fifty steps. VDN-H3 after Stage B is broadly on par with Dense H3 before few-step distillation. Their aesthetic, technical, learned-quality, and FIRM scores are close, although two endpoint-fidelity measures show small degradations. This comparison isolates the hybrid architecture and its adaptation from any reduction in sampling steps.

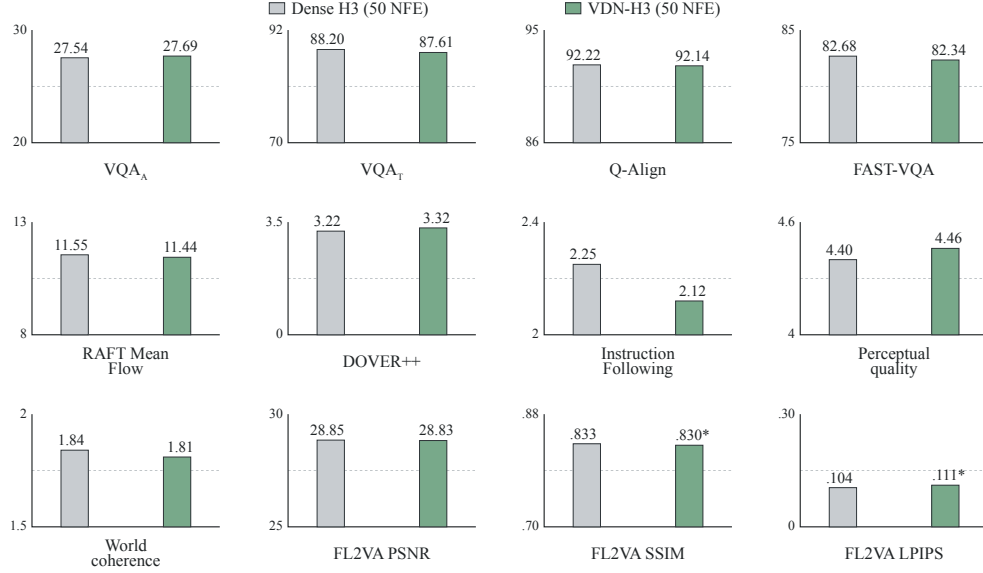


Figure 8: Quality at 50 NFEs before few-step distillation. VDN-H3 is the Stage-B checkpoint.

Eight steps. We compare distilled VDN-H3 with Dense H3 plus Larry’s eight-step adapter and FastH3 v2 at the same NFE count. VDN-H3 scores highest on all five no-reference quality measures and remains close to the dense-plus-Larry reference on FIRM and FL2VA. Its VQA_T score is 10.78 points higher than FastH3 v2. FL2VA scores for FastH3 v2 were not available.

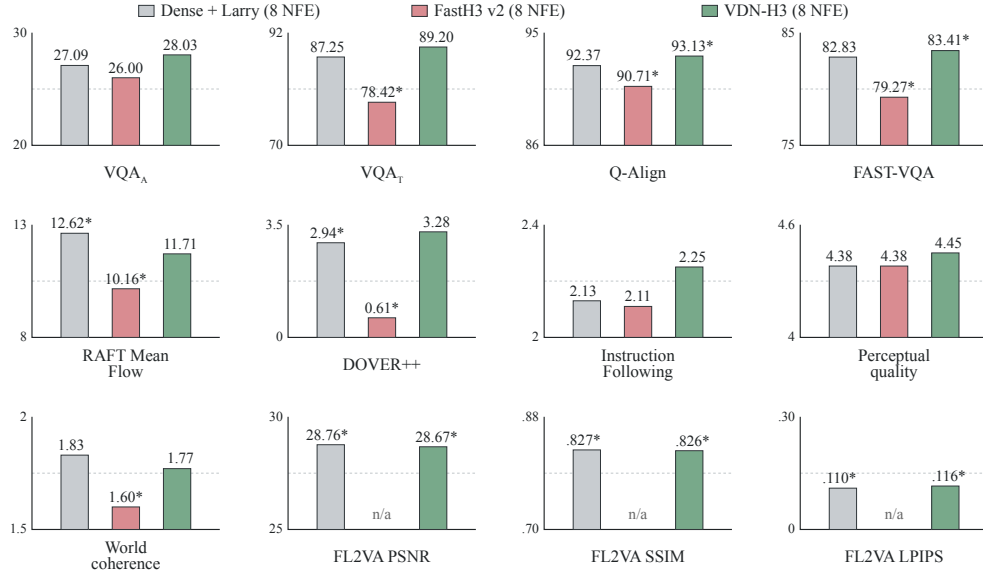


Figure 9: Quality at eight NFEs. FastH3 v2 has no reported FL2VA scores in this comparison.

Four steps. VDN-H3 remains compatible with Larry-style few-step adaptation at the lower sampling budget. It scores above both Dense H3 plus Larry and FastH3 v1 on all five no-reference quality measures; against FastH3 v1, its VQA_T score is higher by 12.64 points, with better FL2VA endpoint fidelity. These are score differences: the supplied significance flags compare each variant with Dense 50, not VDN-H3 directly with FastH3.

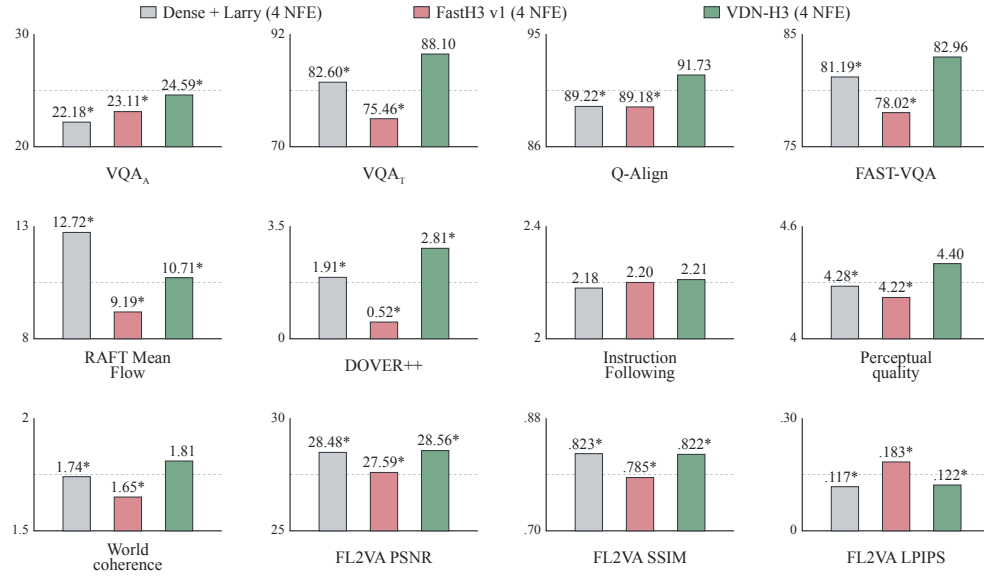


Figure 10: Quality at four NFEs. VDN-H3 retains a clear margin over FastH3 v1 while using the same number of denoising steps.