Attend Locally, Remember Linearly: Linear Attention as Cross-Frame Memory for Autoregressive Video Diffusion

Kunyang Li **Mubarak Shah** **Yuzhang Shang**
Institute of Artificial Intelligence, University of Central Florida

Project Page **Code**

Abstract

Autoregressive (AR) video diffusion is a powerful paradigm for streaming and interactive video generation. However, its reliance on softmax self-attention leads to quadratic compute complexity in sequence length and increasing memory usage due to key-value caching, which fundamentally limits its scalability to long video horizons. Existing remedies (e.g., sparse attention and KV-cache compression) reduce per-step cost but still rely on a linearly growing cache or irreversibly discard past context, and thus fail to simultaneously address linear memory growth and streaming context management. To address this scalability bottleneck, we propose ARL^2 *(**A**ttend **L**ocally, **R**emember **L**inearly)*, a hybrid attention module that replaces quadratic cross-frame attention with a fixed-size recurrent state. Specifically, we decompose self-attention into two branches: an *intra-frame* softmax branch for spatial detail and local dependencies, and an *inter-frame* gated recurrent linear branch that maintains a fixed-size state for streaming context. Our key insight is that softmax attention captures fine-grained local interactions, while a recurrent state provides a controllable mechanism for long-range memory. This design achieves linear-time scaling with constant memory while improving temporal consistency over the full-softmax model. To prevent noisy intermediate states from corrupting long-range memory, we update the recurrent state only after the denoised pass. To avoid within-frame information asymmetry, all tokens share the same pre-update state rather than sequential updates. To the best of our knowledge, this is the first work to convert a pretrained AR video diffusion model into a hybrid linear attention architecture, through an efficient two-stage training scheme adapted to the AR video setting. With 75% of layers replaced by hybrid linear attention, the model achieves up to $2.26\times$ wall-clock speedup and 54% memory reduction, while maintaining comparable quality with improving temporal consistency.

1 Introduction

Autoregressive video diffusion has emerged as a powerful paradigm for streaming and interactive video synthesis, owing to its causal, chunk-wise formulation that naturally aligns with the temporal structure of video and enables streaming inference via key-value caching [Chen et al., 2024, Huang et al., 2025a, Zhu et al., 2026, Yin et al., 2025, Cui et al., 2026, Liu et al., 2026a,b, Yang et al., 2026, Sand.ai et al., 2025]. However, all these systems employ softmax self-attention [Vaswani et al., 2017] in Diffusion Transformers [Peebles and Xie, 2023], whose $O(N^2)$ compute and $O(N)$ memory scaling in sequence length N introduces a structural bottleneck that worsens with every generated block [Xu et al., 2026, Lv et al., 2026]. This cost is already severe at second-scale durations: the KV cache of a 5-second 480p video can exceed 34 GB, surpassing the model parameters themselves [Xi et al., 2026]; and attention accounts for $\sim$75% of total latency after only 14 chunks [Lv et al., 2026], making it the dominant barrier to practical streaming deployment [Yang et al., 2026].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

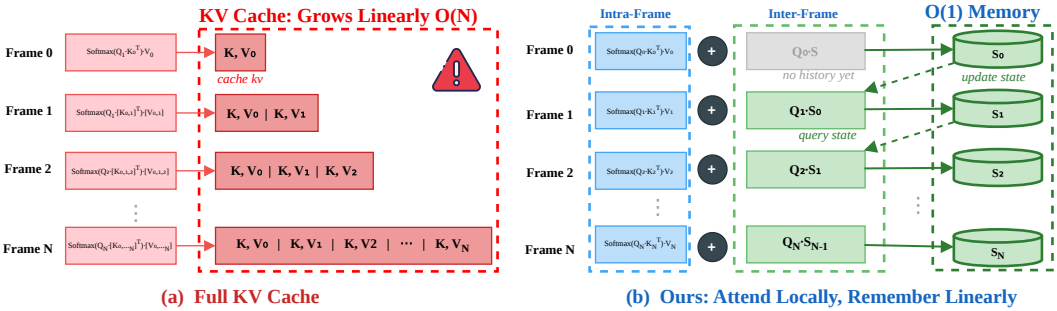


Figure 1: (a) AR video diffusion relies on softmax self-attention with a growing KV cache, incurring $O(N)$ memory, which limits scalability. (b) Replacing self-attention with our hybrid linear attention, the model maintains a fixed-size recurrent state for streaming context with $O(1)$ memory.

Previous efforts have primarily explored sparse attention [Lv et al., 2026, Guo et al., 2026, Huang et al., 2025a, Yang et al., 2026, Liu et al., 2026a], KV cache quantization and eviction [Xi et al., 2026, Chen et al., 2026a, Samuel et al., 2026, Li et al., 2026a] to mitigate this cost. However, these methods have inherent limitations: sparse attention reduces per-step FLOPs but still relies on a linearly growing KV cache; quantization reduces per-entry precision but retains $O(N)$ storage; and eviction bounds memory usage at the cost of irreversible loss of previous context. Consequently, these methods do not simultaneously address the $O(N)$ memory scaling and streaming context management.

These limitations motivate rethinking the problem from a different modeling perspective. Recent work in language modeling reinterprets linear attention as a form of associative memory [Schlag et al., 2021], with modern variants introducing gating and error-correction mechanisms for more precise and stable updates [Yang et al., 2024a,b, 2025a]. Building on this perspective, recent approaches replace partial softmax attention layers with linear attention via distillation or retraining, improving long-context modeling and outperforming either paradigm alone [De et al., 2024, Chen et al., 2026b].

However, directly transferring these advances to AR video diffusion is non-trivial. While linear attention can serve as a recurrent memory for long-range context in LLMs, no prior work has explored how to convert a pretrained AR video diffusion model into such a hybrid architecture. Unlike the homogeneous causal attention used in LLMs, AR video diffusion exhibits an inherently heterogeneous attention structure: intra-frame bidirectional attention captures rich spatial correlations [Fan et al., 2025], while inter-frame causal attention relies on a growing KV cache to maintain temporal consistency [Sand.ai et al., 2025]. This heterogeneity introduces two key challenges absent in the LLM setting: **(1)** how to design a hybrid attention module that respects the distinct roles of intra- and inter-frame attention; **(2)** how to maintain a stable recurrent state across frame-level diffusion denoising.

We address ❶ challenge with a role-aware decomposition of attention into intra-frame and inter-frame branches. Inter-frame causal attention maintains temporal consistency over a growing context, while intra-frame bidirectional attention models fine-grained spatial structure (Fig. 1). Accordingly, we replace the KV cache with a fixed-size recurrent state via a Gated Delta Network [Yang et al., 2025a] (*remember linearly*), whose gated forgetting and delta update rules enable controllable long-range memory management at constant cost, while retaining intra-frame softmax attention to preserve spatial fidelity (*attend locally*; Fig. 2). Since all tokens within a frame are decoded jointly, retaining intra-frame softmax does not introduce additional latency.

The ❷ challenge arises from the interaction between recurrent state and frame-level diffusion denoising. Sequential state updates would introduce intra-frame information asymmetry, and repeated exposure to noisy intermediates can corrupt the accumulated memory over long horizons. To address this, we introduce two mechanisms (Fig. 3c): all tokens share a pre-update state to ensure consistent context, and the recurrent state is updated only after the final clean pass to prevent contamination. Ablations (Sec. 4) show that they are critical for maintaining video quality.

To realize this design in practice, we propose a progressive two-stage distillation pipeline to efficiently and stably convert a pretrained Causal Forcing model [Zhu et al., 2026] into the hybrid architecture. Stage 1 performs per-layer distillation employing sensitivity-guided layer selection adapted to the AR video setting to minimize performance degradation, while Stage 2 conducts joint distillation to train the hybrid model end-to-end. With 50% of attention layers replaced by the hybrid linear

attention, the hybrid model achieves up to $1.57\times$ wall-clock speedup and 35% memory reduction, while maintaining comparable quality and improving temporal consistency with Temporal Flickering and Motion Smoothness , compared to the full-softmax teacher. At 75% replacement, efficiency further improves to $2.26\times$ speedup and 54% memory reduction.

2 Related work

Autoregressive video diffusion. Representative open-source video Diffusion Transformers (DiTs), such as Wan [Team Wan et al., 2025], CogVideoX [Yang et al., 2025b], and Open-Sora [Zheng et al., 2024a], rely on spatiotemporal self-attention, leading to quadratic scaling in compute and memory with video length. To enable streaming generation, a recent line of work reformulates video diffusion as frame-autoregressive (AR): Diffusion Forcing [Chen et al., 2024] assigns independent noise levels per token, bridging AR and full-sequence diffusion; Self-Forcing [Huang et al., 2025a] closes the train-test exposure gap by conditioning on self-generated context during training, achieving real-time streaming; and Causal Forcing [Zhu et al., 2026] further improves quality through ODE distillation with an AR teacher. However, all these AR approaches are bottlenecked by linearly growing KV cache. We propose a hybrid attention module for autoregressive video diffusion that replaces inter-frame softmax attention with a fixed-size recurrent state and retains intra-frame bidirectional softmax interactions, achieving $O(1)$ memory per-layer with respect to sequence length.

Linear attention for video. Existing efforts to incorporating linear attention into video generation can be broadly categorized as follows. *Full-architecture retraining.* SANA-Video [Chen et al., 2026c] adopts block-linear attention with constant-memory states, while SSM-based models explore alternative sequence modeling paradigms for long-horizon video diffusion and world models [Oshima et al., 2024, Hong et al., 2025, Po et al., 2025]. Although effective, these methods require large-scale retraining. *Supplementary branch.* SALAD [Fang et al., 2026] introduces a lightweight linear attention branch alongside sparse softmax attention, and VideoSSM [Yu et al., 2025] combines sliding-window softmax with an SSM-based global memory. In both cases, linear-complexity modules serve as auxiliary components, while the primary softmax path and its associated complexity remain. *Attention replacement in bidirectional DiTs.* Attention Surgery [Ghafoorian et al., 2025] and SLA [Zhang et al., 2026] replace selected attention blocks in pretrained video DiTs with kernel-based or sparse-linear approximations, achieving efficiency gains with minimal fine-tuning. However, they operate within the bidirectional denoising paradigm, where each step processes the full spatiotemporal sequence, so memory and computation still scale with video length and do not support streaming generation. ReHyAt [Ghafoorian and Habibian, 2026] introduces a recurrent hybrid attention formulation for chunk-wise inference with constant memory, but is distilled from bidirectional diffusion models that assume full-sequence availability, leading to a mismatch for causal generation. *In contrast*, we convert a pretrained AR video diffusion model into a hybrid architecture with gated recurrent linear attention using a Causal Forcing teacher to preserve causal attention behavior. This is the first work to extend hybrid-attention distillation to the AR setting, enabling streaming video generation with constant memory without retraining from scratch.

3 Method

We first review the attention mechanism in AR video diffusion and its memory bottleneck (Sec. 3.1). We then introduce ARL^2, a hybrid linear attention module that decomposes self-attention into intra-frame softmax and inter-frame recurrent branches (Sec. 3.2). Next, we describe two state management strategies: frame-level access for consistent history observation and clean-pass-only updates to prevent contamination from noisy intermediates (Sec. 3.3). Finally, we present a lightweight two-stage distillation pipeline that converts a pretrained AR model into the hybrid architecture (Sec. 3.4).

3.1 Preliminaries

The autoregressive video diffusion model [Chen et al., 2024, Huang et al., 2025a, Zhu et al., 2026] generates video chunk-by-chunk in an autoregressive manner, where each chunk contains one or more contiguous frames. Let $x_N \in \mathbb{R}^{L\times d}$ denote the L latent tokens of frame N, where d is the hidden dimension. At each self-attention layer, the model computes queries, keys, and values $Q = xW_q$,

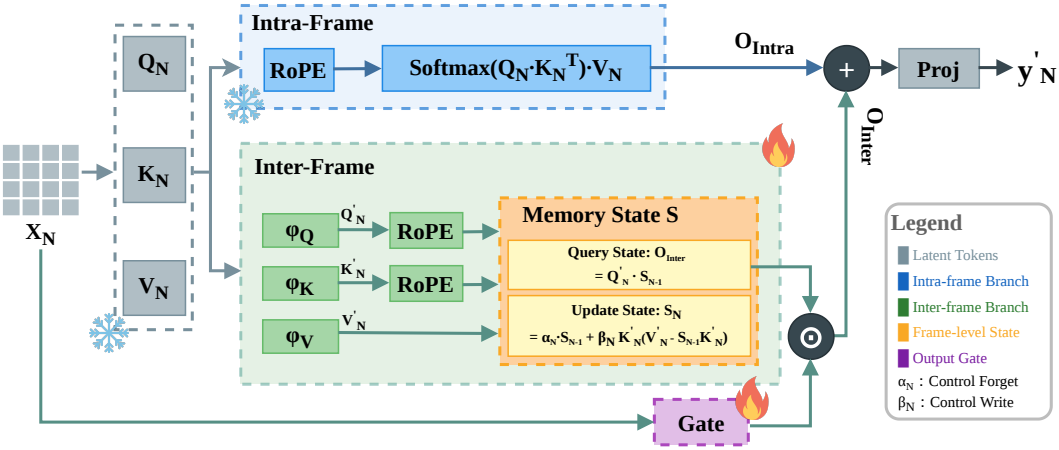


Figure 2: ARL^2 attention module (normalization omitted for clarity). Given tokens X_N of frame N, Q_N, K_N, V_N, which are routed to two branches. The intra-frame branch (top) applies bidirectional softmax attention over tokens within the current frame, producing O_{intra}. The inter-frame branch (bottom) applies recurrent linear attention, where a fixed-size state S maintains and updates long-range memory across frames, and produces O_{inter}. The state S is updated once per frame. During training, the hybrid module is learned via a progressive two-stage distillation pipeline (See Sec. 3.4).

$K = xW_k$, $V = xW_v$ for the current frame, and concatenates keys and values from previous frames into the KV cache. The attention output over the full context is:

$$y = \text{softmax}\left(\frac{Q\,K^\top}{\sqrt{d_k}} + M\right) V, \tag{1}$$

where the causal block mask M permits bidirectional attention within the current frame and causal attention to previous frames. As video length grows, the KV cache accumulates linearly: generating frame N requires storing $(N-1)\times L$ key-value pairs per layer, making memory scale as $\mathcal{O}(N)$.

3.2 ARL^2: Attend Locally, Remember Linearly

Our key insight stems from the heterogeneous attention structure of autoregressive video diffusion: intra-frame attention is bidirectional, modeling fine-grained spatial correlations, while inter-frame attention is causal and relies on a growing KV cache to maintain temporal consistency. Exploiting this asymmetry, we decompose the original self-attention into two branches (Fig. 2). One branch preserves the original intra-frame softmax attention, reusing the frozen pretrained projections:

$$O_{\text{intra}} = \text{Softmax}(\text{RoPE}(Q), \text{RoPE}(K), V), \tag{2}$$

The other branch replaces the inter-frame KV cache with a recurrent linear state $S \in \mathbb{R}^{H\times D\times D}$. Because the pretrained Q, K, V are optimized for softmax attention, we introduce per-head grouped 1×1 feature maps ϕ_q, ϕ_k, ϕ_v to adapt the representations to the linear recurrence domain:

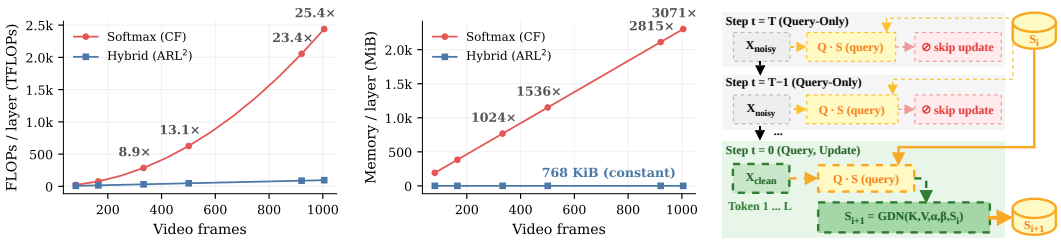
$$Q' = \text{L2Norm}(\text{RoPE}(\phi_q(Q))), \quad K' = \text{L2Norm}(\text{RoPE}(\phi_k(K))), \quad V' = \phi_v(V), \tag{3}$$

We build our linear attention module on the Gated Delta Network [Yang et al., 2025a], whose state update rule is:

$$S_N = \alpha_N\, S_{N-1} + \beta_N\, K'^{\top}_N (V'_N - K'_N\, S_{N-1})\,, \tag{4}$$

where α_N (forget gate) and β_N (learning rate) are produced by per-token projections of the hidden state. Eq. 4 uses frame-level notation for brevity; in practice the update is applied sequentially over all L tokens within the frame via an efficient chunkwise kernel [Yang et al., 2024a], with per-token gates $\alpha^{(j)}, \beta^{(j)}$ for $j \in \{1, \ldots, L\}$. When generating frame N, the inter-frame output is obtained by querying the state conditioned on all previous frames:

$$O_{\text{inter}} = Q'_N\, S_{N-1}, \tag{5}$$



(a) Theoretical per-layer FLOPs. (b) Per-layer memory scaling. (c) State management.

Figure 3: (a) Softmax attention scales quadratically with video length, while the hybrid layer scales linearly. (b) The KV cache grows linearly with video length, whereas ours remains constant. (c) The recurrent state S is shared across all tokens within a frame and updated only after the final clean pass.

where all tokens in the current frame query the same pre-update state S_{N-1}. Then two branches are combined via a lightweight headwise gate:

$$G = \sigma(x\, W_g), \tag{6}$$

$$y' = (O_{\text{intra}} + G \odot O_{\text{inter}})\, W_o. \tag{7}$$

where $W_g \in \mathbb{R}^{d\times H}$ projects to one scalar per head, σ denotes the sigmoid function, and $G \in [0,1]^{L\times H}$ is broadcast over the head dimension D before the element-wise product with $O_{\text{inter}} \in \mathbb{R}^{L\times H\times D}$.

The fixed-size state $S \in \mathbb{R}^{H\times D\times D}$ reduces cumulative per-layer FLOPs from quadratic to linear scaling and keeps per-layer memory constant with respect to the number of generated frames (Fig. 3a, 3b). In practice, replacing 75% of layers yields up to 2.26× wall-clock speedup and 54% memory reduction (Sec. 4.2).

3.3 Memory state management for AR video diffusion

Recurrent linear attention in LLMs [Yang et al., 2024a,b, 2025a] updates the state at each token sequentially. Directly applying this to AR video diffusion introduces two issues: (1) tokens within a frame should share the same cross-frame context, but token-level updates cause asymmetry; (2) multi-step denoising writes noisy intermediates into the state, corrupting accumulated memory. We address both with two complementary mechanisms (Fig. 3c).

Frame-level access. All L tokens within a frame query the same pre-update state S_{N-1}, and the state is updated only after the entire frame has been processed:

$$O^{(i)}_{\text{inter},N} = Q'^{(i)}_N\, S_{N-1}, \quad \forall\, i \in \{1,\dots,L\}. \tag{8}$$

Clean-pass update. During the multi-step denoising loop, the model queries S but does not update it. Once a clean frame is produced, its tokens update S via Eq. 4. This also reduces the number of state writes from one per denoising step to one per frame, improving inference efficiency.

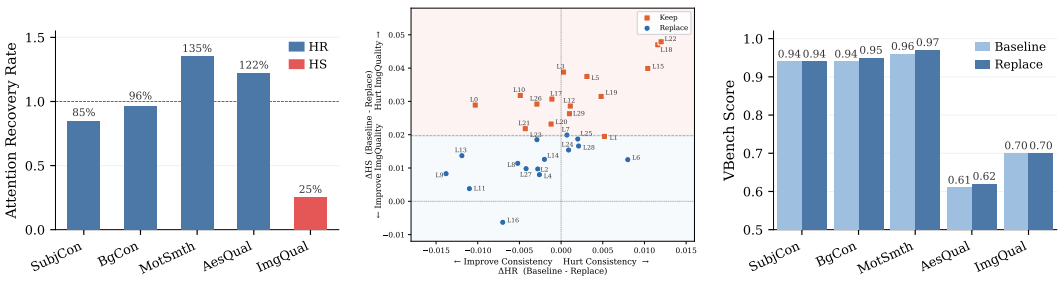
3.4 Training pipeline: Two-stage distillation

Rather than retraining the full model, we selectively replace a subset of attention layers and distill them in a progressive two-stage pipeline: Stage 1 independently distills each attention layer and select which layers to replace; Stage 2 jointly fine-tunes the hybrid model so that all replaced layers operate coherently. This yields strong performance with modest compute (Sec. 4), consistent with findings on progressive distillation for hybrid LLMs [Chen et al., 2026b, Li et al., 2026b].

Stage 1: per-layer distillation and selection. We distill all 30 attention layers independently. For each layer ℓ, the teacher attention and student hybrid module run in parallel on the teacher hidden state, and we minimize their output MSE:

$$\mathcal{L}^{(\ell)}_{\text{align}} = \frac{1}{L\cdot d}\, \|y'_\ell(x_t,t,c) - y_\ell(x_t,t,c)\|^2_F\,, \tag{9}$$

where y_ℓ and y'_ℓ denote the teacher and student outputs, x_t is the noisy latent at timestep t, and c is the text condition. The teacher output is propagated to subsequent layers, ensuring distillation



(a) Per-dimension ARR analysis. (b) HS-protected layer selection. (c) End-to-end validation.

Figure 4: (a) Per-dimension ARR averaged across layers. Four Hybrid-Recoverable (HR) dimensions achieve $\geq$85% recovery, while ImgQual is Hybrid-Sensitive (HS) at $\sim$25%, motivating HS-aware layer selection to minimize quality degradation. (b) Each point corresponds to one of the 30 layers, plotted by ΔHR (x-axis) and ΔHS (y-axis). Here the 50% lowest-p_ℓ layers (blue) are selected for replacement. (c) End-to-end validation after Stage 2: the hybrid model matches or exceeds the baseline on all HR dimensions and recovers ImgQual to within $\sim$1%, indicating that HS-aware selection preserves sensitive dimensions while allowing HR gaps to be closed via joint distillation.

under the intact teacher context. Only the state-related parameters are trained, including feature maps ϕ_q, ϕ_k, ϕ_v (Eq. 3), the headwise gate W_g (Eq. 6), and the state update projections for α_N, β_N (Eq. 4). Training uses text-conditioned inputs with randomly sampled noisy latents. After per-layer distillation, we determine which layers to replace, in three steps:

(i) Layer analysis. For each layer we evaluate three conditions on our stress-test prompts and measure VBench scores across 5 quality dimensions (indexed by i): (1) **Baseline**$_i$: original softmax with full KV cache; (2) **Replace**$_{\ell,i}$: layer ℓ alone replaced with its Stage-1 hybrid module; (3) **Skip**$_{\ell,i}$: attention output of layer ℓ zeroed via a modulated skip connection. From these VBench scores we compute the Attention Recovery Rate (ARR) per layer ℓ and quality dimension i:

$$\mathrm{ARR}_{\ell,i} = \frac{\mathrm{Replace}_{\ell,i} - \mathrm{Skip}_{\ell,i}}{\mathrm{Baseline}_i - \mathrm{Skip}_{\ell,i}}, \tag{10}$$

where ARR=1 denotes full recovery of the softmax attention's contribution to dimension i. Aggregating ARR across layers (Fig. 4a splits the five dimensions into two groups. **Hybrid-Recoverable (HR)**: SubjCon, BgCon, MotSmth, AesQual, where hybrid module achieves $\geq$85% recovery. **Hybrid-Sensitive (HS)**: ImgQual, where hybrid module recovers only $\sim$25% of the attention contribution.

(ii) Layer selection. Motivated by the hypothesis that HS dimensions are intrinsically sensitive to attention replacement, we design layer selection to explicitly safeguard them. Concretely, each layer is scored by its impact on HS, with a small HR penalty acting as a tiebreaker:

$$p_\ell \;=\; \Delta\mathrm{HS}_\ell \;+\; \beta \cdot \max\big(\Delta\mathrm{HR}_\ell,\, 0\big), \tag{11}$$

where $\Delta\mathrm{HS}_\ell$ and $\Delta\mathrm{HR}_\ell$ denote the mean degradation across HS and HR dimensions, respectively. Under a 50% layer replacement setting, we replace the 15 layers with the smallest p_ℓ (Fig. 4b). Sweeping $\beta \in \{0.1, 0.5, 1.0\}$ yields the identical replacement set, so β functions as a boundary-layer tiebreaker rather than a sensitive hyperparameter.

(iii) End-to-end validation. After Stage 2 joint distillation, the resulting hybrid model matches or exceeds the softmax baseline on all four HR dimensions and recovers ImgQual to within $\sim$1% on the same stress-test prompts (Fig. 4c). These results indicate that the HS-protected layer selection effectively prevents severe degradation in hybrid-sensitive layers, while allowing residual errors in hybrid-recoverable layers to be compensated during joint training.

Stage 2: joint distillation. The student is initialized from the teacher checkpoint, the Stage 1 GDN weights are loaded into the selected layers, and the resulting hybrid model is jointly distilled by matching the teacher's predicted velocity field. Let $v_\theta(x_t, t, c)$ denote the flow velocity predicted by the diffusion model given noisy latent x_t, timestep t, and text condition c. The joint loss is:

$$\mathcal{L}_{\text{joint}} = \left\| v_\theta^{\text{student}}(x_t, t, c) - v_\theta^{\text{teacher}}(x_t, t, c) \right\|_2^2 . \tag{12}$$

In this stage, the teacher is frozen, while the student trains the hybrid linear attention module parameters, LoRA adapters on the Q, K, V path, and the FFN of each selected block.

Table 1: ARL^2 achieves comparable overall performance to representative open-source video models, while improving temporal flickering and background consistency, with $\mathcal{O}(1)$ memory complexity. ARL^2 (50%/75%) denotes replacing 50%/75% of attention layers with hybrid linear attention.

Model	#Params	Subject Consistency	Background Consistency	Temporal Flickering	Motion Smoothness	Aesthetic Quality	Imaging Quality	**Quality Avg.↑**	**Semantic Avg.↑**
Bidirectional Diffusion									
LTX-Video [HaCohen et al., 2024]	1.9B	87.89	92.00	98.20	97.91	48.08	58.62	80.45	51.32
Wan2.1-T2V [Team Wan et al., 2025]	1.3B	96.18	98.33	98.33	96.60	65.35	68.64	87.24	63.01
Linear Diffusion									
SANA-Video [Chen et al., 2026c]	2B	97.09	96.95	89.22	95.10	70.48	67.55	86.07	**80.76**
Autoregressive Diffusion									
SkyReels-V2 [Chen et al., 2025]†	1.3B	97.78	97.68	97.97	98.50	59.66	62.29	85.65	48.02
MAGI-1 [Sand.ai et al., 2025]	4.5B	95.71	97.14	97.69	98.67	62.44	64.84	86.08	72.02
Distilled Autoregressive									
Self Forcing [Huang et al., 2025a]	1.3B	95.62	96.40	96.82	**95.59**	66.75	69.96	86.86	70.71
Causal Forcing [Zhu et al., 2026]	1.3B	**96.19**	**96.44**	96.27	94.24	**68.80**	**70.62**	87.09	70.97
Distilled Linear Autoregressive									
ARL^2 (50%)	1.3B	95.63	96.41	**97.26**	**95.59**	68.29	69.87	**87.17**	70.13
ARL^2 (75%)	1.3B	94.85	95.94	96.90	94.24	68.44	69.87	86.71	70.24

4 Experiments

4.1 Setup

Training. We adopt Causal Forcing [Zhu et al., 2026] as base model. In Stage 1, all 30 layers are independently distilled for 6K steps using prompts from VidProM [Wang and Yang, 2024]; then identifies 15 layers (50%) or 23 layers (75%) for replacement. Stage 2 jointly distills the hybrid student against the frozen Causal Forcing teacher for 12K steps on 6K teacher-synthesized videos. The entire pipeline trains fewer than 2% of backbone parameters, consumes ∼156 H100 GPU-hours (∼50 for Stage 1, ∼106 for Stage 2).

Evaluation. We use VBench [Huang et al., 2024] as the primary benchmark. All models are evaluated at 832×480 with 81 frames (†: native resolution 544×960, 97 frames). Layer sensitivity analysis uses a separate set of stress-test prompts designed to probe temporal consistency and visual fidelity under challenging conditions (details in App. C). For efficiency, we evaluate end-to-end generation time, throughput (FPS), and peak GPU memory on a subset of prompts, measured on a single NVIDIA RTX PRO 6000 Blackwell under cold-start conditions, across video lengths from 81 to 921 frames.

Table 2: ARL^2 achieves consistent efficiency gains with up to 2.26× speedup and significantly reduced memory, while maintaining competitive or improved video quality.

Video Frames	Model	*Efficiency* Gen (s)↓	FPS ↑	Mem (GB)↓	Speed-up↑	*Quality*↑ Qual. Avg	Sem. Avg	Total
81	Self Forcing	8.70	9.31	23.30	1.00×	86.79	69.15	83.26
	Causal Forcing	8.68	9.33	23.30	1.00×	86.69	69.11	83.17
	ARL^2 (50%)	8.70	9.31	20.53	1.00×	**87.81**	**71.41**	**84.53**
	ARL^2 (75%)	**8.67**	**9.34**	**19.10**	1.00×	86.00	65.86	81.97
165	Self Forcing	22.27	7.41	29.13	1.00×	86.01	66.25	82.05
	Causal Forcing	22.28	7.41	29.13	1.00×	85.48	70.55	82.49
	ARL^2 (50%)	19.90	8.29	23.54	1.12×	**86.32**	68.65	**82.79**
	ARL^2 (75%)	**18.63**	**8.86**	**20.60**	**1.20**×	84.36	**71.75**	81.84
501	Self Forcing	124.24	4.03	54.08	1.00×	85.14	53.94	78.90
	Causal Forcing	124.24	4.03	54.08	1.00×	83.13	**58.96**	78.30
	ARL^2 (50%)	88.52	5.66	37.26	1.40×	**86.04**	58.85	**80.60**
	ARL^2 (75%)	**71.54**	**7.00**	**28.30**	**1.74**×	77.00	55.35	72.67
921	Self Forcing	357.74	2.57	87.88	1.00×	83.37	**67.83**	**80.26**
	Causal Forcing	357.52	2.58	87.88	1.00×	79.96	58.43	75.66
	ARL^2 (50%)	227.48	4.05	57.00	1.57×	**84.96**	51.62	78.29
	ARL^2 (75%)	**158.16**	**5.82**	**40.50**	**2.26**×	77.34	48.40	71.55

4.2 Results

Quality. As shown in Tab. 1, we compare ARL^2 with representative open-source video generation models, including LTX-Video [HaCohen et al., 2024], Wan2.1 [Team Wan et al., 2025], SANA-Video [Chen et al., 2026c], SkyReels-V2 [Chen et al., 2025], MAGI-1 [Sand.ai et al., 2025]), and Self Forcing [Huang et al., 2025a], Causal Forcing [Zhu et al., 2026]).

ARL^2 (50%) achieves the higher Quality score among all distilled models (87.17), with improvements in temporal flickering (+0.99) and motion smoothness (+1.35), suggesting improved consistency. ARL^2 achieves comparable quality to the retrained

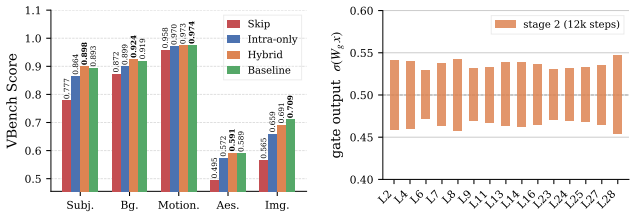
SANA-Video (linear model), while matching the much larger MAGI-1 (4.5B, 86.08) with 3.5× fewer parameters. Details (App. E) show gains in object recognition (+1.10) and color fidelity (+0.76), but weaker performance on spatial relationships and human actions, suggesting limits of a fixed-size state for precise spatial modeling.

Efficiency. Tab. 2 highlights our computational and memory efficiency through wall-clock time and peak memory across increasing video lengths. Speedup increases with video length, consistent with the linear–quadratic scaling gap in Fig. 3a. ARL^2 (75%) reaches 1.20×, 1.74×, and 2.26× at 165, 501, and 921 frames, respectively. Even ARL^2 (50%) achieves 1.57× at 921 frames, confirming that gains grow as KV-cache overhead dominates.

For memory, hybrid layers replace the $O(N)$ KV-cache with a fixed-size state, reducing per-frame memory from 293 MB to 175 MB (40%). At 921 frames, ARL^2 (75%) uses 40.5 GB vs. 87.9 GB for Causal Forcing (54%). At 1005 frames, Causal Forcing runs out of memory (> 91 GB), while ARL^2 completes in 179 s using 43 GB, demonstrating scalability. All results use batch size 1; the memory savings enable higher parallelism and throughput. Beyond efficiency, ARL^2 (50%) maintains strong quality as video length increases. This suggests the recurrent state provides more stable long-range context than the KV-cache, consistent with improved inter-frame consistency in Tab. 1.

4.3 Ablation study

We ablate two core design choices using 6 hybrid layers $[2, 4, 7, 8, 23, 24]$ trained with Stage 1 and Stage 2 (3K steps each), evaluated on stress-test prompts at 21 frames.



(a) Attention decomposition. (b) Gate activation.

Figure 5: (a) Attention decomposition on a representative layer (23), highlighting complementary intra- and inter-frame branch modeling. (b) Gate activation across hybrid layers after Stage 2 training, indicating that both branches remain active and are combined via a content-dependent mixture.

Branch contribution. To isolate intra- and inter-frame roles, we evaluate four settings on a single hybrid layer: *Skip*, *Intra-only*, *Hybrid*, and *Baseline*. As shown in Fig. 5a (layer 23 as an example), removing the layer (*Skip*) degrades all metrics, indicating replaced layers remain essential. The intra-frame branch alone ((*Intra-only*)) largely recovers subject and background consistency, highlighting its role in local spatial reasoning. Adding the recurrent branch (*Hybrid*) further improves background consistency and aesthetics, and slightly surpasses the teacher (*Baseline*) on motion smoothness, suggesting complementary temporal modeling. Fig. 5b shows the learned gate $\sigma(W_g x)$ stays in $[0.45, 0.55]$ with per-head variation, indicating both branches remain active and form a content-dependent mixture.

State access granularity. We compare frame-level access (all tokens query the same S_{N-1}) against token-level access (each token sees its own sequentially accumulated state). Frame-level reaches Avg 83.3, matching the full-softmax teacher (83.2), while token-level drops to 82.7 (Tab. 3). This confirms that frame-level access is necessary to avoid intra-frame context asymmetry introduced by token-level updates.

Table 3: Frame-level is superior to token-level access.

Setting	SubjCon	BgCon	MotSmth	AesQual	ImgQual	Avg
Causal Forcing	**93.7**	93.7	96.5	**61.3**	**70.9**	83.2
Token-Level	93.2	94.2	96.7	60.6	69.0	82.7
Frame-Level	93.6	**94.3**	**97.4**	61.1	70.3	**83.3**

Gate granularity. The branch-fusion gate $\sigma(W_g x)$ can be parameterized as scalar (1.5K params per layer), headwise (18K), or elementwise (2.36M). Elementwise gating is strictly worse (−1.4), due to poor convergence under its 130× larger parameterization with limited training steps. Scalar and headwise both match or exceed the Causal Forcing teacher, with headwise achieving the highest Avg (83.5) while providing per-head adaptability at modest cost.

Table 4: Headwise gating achieves the best trade-off between performance and parameter efficiency.

Setting	Params	SubjCon	BgCon	MotSmth	AesQual	ImgQual	Avg
Elementwise	2.36M	92.6	93.8	96.1	60.0	67.8	82.1
Headwise	18K	**93.8**	**94.5**	**97.6**	61.3	**70.3**	**83.5**
Scalar	1.5K	**93.8**	94.4	97.5	**61.4**	69.6	83.3

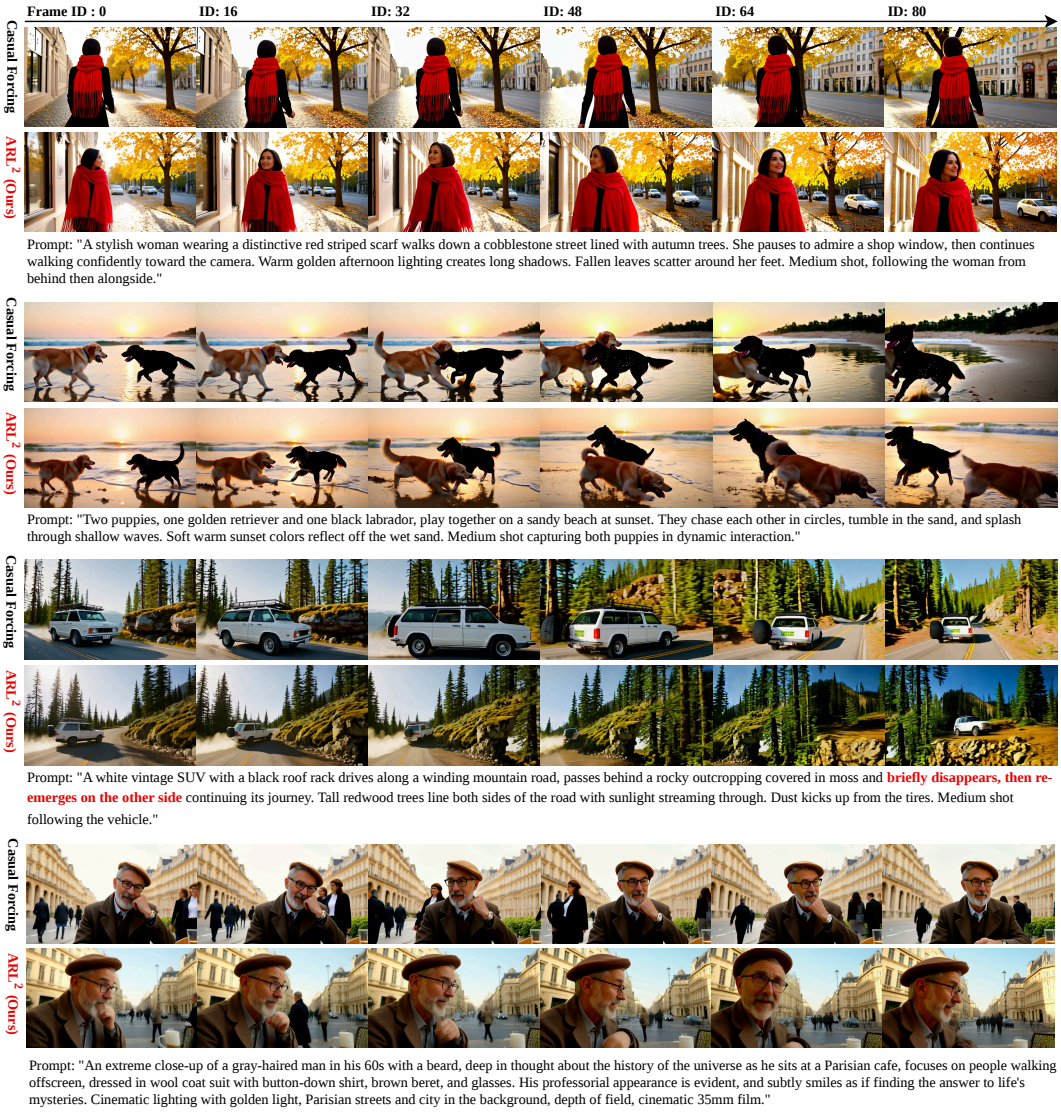


Figure 6: Qualitative comparison. Highlighting strong visual quality and temporal consistency under significant computational and memory efficiency. ARL^2 achieves up to $2.26\times$ speedup and 54% memory reduction with comparable or better quality. In the "white vintage SUV" example, ARL^2 preserves appearance consistency when the car disappears and reappears, while Causal Forcing fails to follow the prompt and produces distortions.

5 Conclusion

We presented ARL^2, a hybrid attention architecture for autoregressive video diffusion that preserves bidirectional intra-frame softmax attention for fine-grained intra-frame spatial modeling while replacing the growing cross-frame KV cache with a fixed-size gated recurrent state. Two complementary mechanisms, frame-level state access and clean-pass updates, ensure stable streaming memory under iterative diffusion denoising. A lightweight two-stage distillation pipeline, guided by sensitivity-aware layer selection, converts a pretrained softmax model into the hybrid architecture with fewer than 2% trainable parameters and $\sim$156 H100 GPU-hours. At 50% layer replacement, ARL^2 matches or exceeds the softmax teacher in overall quality while improving temporal consistency, with $1.57\times$ speedup and 35% memory reduction. At 75% replacement, efficiency scales to $2.26\times$ speedup and 54% memory savings while maintaining competitive generation quality, enabling long-form video synthesis beyond the memory limits of full-attention baselines.

References

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. In *International Conference on Machine Learning (ICML)*, 2026.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22963–22974, 2025.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. In *International Conference on Learning Representations (ICLR)*, 2026.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *International Conference on Learning Representations*, volume 2026, pages 91177–91196, 2026a.

Jinxiu Liu, Xuanming Liu, Kangfu Mei, Yandong Wen, Ming-Hsuan Yang, and Weiyang Liu. Streaming autoregressive video generation via diagonal distillation. In *International Conference on Learning Representations (ICLR)*, 2026b.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Ying-Cong Chen, Yao Lu, Song Han, and Yukang Chen. Longlive: Real-time interactive long video generation. In *International Conference on Learning Representations (ICLR)*, 2026.

Sand.ai, Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, W. Q. Zhang, Weifeng Luo, Xiaoyang Kang, Yuchen Sun, Yue Cao, Yunpeng Huang, Yutong Lin, Yuxin Fang, Zewei Tao, Zheng Zhang, Zhongshu Wang, Zixun Liu, Dai Shi, Guoli Su, Hanwen Sun, Hong Pan, Jie Wang, Jiexin Sheng, Min Cui, Min Hu, Ming Yan, Shucheng Yin, Siran Zhang, Tingting Liu, Xianping Yin, Xiaoyu Yang, Xin Song, Xuan Hu, Yankai Zhang, and Yuqiao Li. Magi-1: Autoregressive video generation at scale. *ArXiv*, abs/2505.13211, 2025. URL https://api.semanticscholar.org/CorpusID:278768832.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:13756489.

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

Yichun Xu, Navjot K. Khaira, and Tejinder Singh. Kv cache optimization strategies for scalable and efficient llm inference. *arXiv preprint arXiv:2603.20397*, 2026.

Chengtao Lv, Yumeng Shi, Yushi Huang, Ruihao Gong, Shen Ren, and Wenya Wang. Light forcing: Accelerating autoregressive video diffusion via sparse attention. In *International Conference on Machine Learning (ICML)*, 2026.

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. In *International Conference on Machine Learning (ICML)*, 2026.

Hang Guo, Zhaoyang Jia, Jiahao Li, Bin Li, Yuanhao Cai, Jiangshan Wang, Yawei Li, and Yan Lu. Efficient autoregressive video diffusion with dummy head. *ArXiv*, abs/2601.20499, 2026. URL https://api.semanticscholar.org/CorpusID:285101812.

Hanmo Chen, Chenghao Xu, Xu Yang, Xuan Chen, and Cheng Deng. Past- and future-informed kv cache policy with salience estimation in autoregressive video diffusion. *ArXiv*, abs/2601.21896, 2026a. URL https://api.semanticscholar.org/CorpusID:285139913.

Dvir Samuel, Issar Tzachor, Matan Levy, Michael Green, Gal Chechik, and Rami Ben-Ari. Fast autoregressive video diffusion and world models with temporal cache compression and sparse attention. In *International Conference on Machine Learning (ICML)*, 2026.

Kunyang Li, Mubarak Shah, and Yuzhang Shang. Packcache: A training-free acceleration method for unified autoregressive video generation via compact kv-cache. *ArXiv*, abs/2601.04359, 2026a. URL https://api.semanticscholar.org/CorpusID:284544089.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, 2021. URL https://api.semanticscholar.org/CorpusID:235377069.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *International Conference on Machine Learning (ICML)*, 2024a.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations (ICLR)*, 2025a.

Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins, Arnaud Doucet, David Budden, Yee Whye Teh, Razvan Pascanu, Nando de Freitas, and Caglar Gulcehre. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *ArXiv*, abs/2402.19427, 2024. URL https://api.semanticscholar.org/CorpusID:268091246.

Yingfa Chen, Zhen Leng Thai, Zihan Zhou, Zhu Zhang, Xingyu Shen, Shuo Wang, Chaojun Xiao, Xu Han, and Zhiyuan Liu. Hybrid linear attention done right: Efficient distillation and effective architectures for extremely long contexts. *arXiv preprint arXiv:2601.22156*, 2026b.

Qihang Fan, Huaibo Huang, and Ran He. Breaking the low-rank dilemma of linear attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25271–25280, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *ArXiv*, abs/2503.20314, 2025. URL https://api.semanticscholar.org/CorpusID:277321639.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations (ICLR)*, 2025b.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *ArXiv*, abs/2412.20404, 2024a. URL https://api.semanticscholar.org/CorpusID:275133398.

Junsong Chen, Yuyang Zhao, Jincheng Yu, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Daquan Zhou, Huan Ling, Haozhe Liu, Hongwei Yi, Hao Zhang, Muyang Li, Yukang Chen, Han Cai, Sanja Fidler, Ping Luo, Song Han, and Enze Xie. Sana-video: Efficient video generation with block linear diffusion transformer. In *International Conference on Learning Representations (ICLR)*, 2026c.

Yuta Oshima, Shohei Taniguchi, Masahiro Suzuki, and Yutaka Matsuo. Ssm meets video diffusion models: Efficient long-term video generation with structured state spaces. *arXiv preprint arXiv:2403.07711*, 2024.

Yicong Hong, Long Mai, Yuan Yao, and Feng Liu. Pushing the boundaries of state space models for image and video generation. *ArXiv*, abs/2502.00972, 2025. URL https://api.semanticscholar.org/CorpusID:276094439.

Ryan Po, Yotam Nitzan, Richard Zhang, Berlin Chen, Tri Dao, Eli Shechtman, Gordon Wetzstein, and Xun Huang. Long-context state-space video world models. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8733–8744, 2025. URL https://api.semanticscholar.org/CorpusID:278911218.

Tongcheng Fang, Hanling Zhang, Ruiqi Xie, Zhuo Han, Xin Tao, Tianchen Zhao, Pengfei Wan, Wenbo Ding, Wanli Ouyang, Xuefei Ning, and Yu Wang. Salad: Achieve high-sparsity attention via efficient linear attention tuning for video diffusion transformer. *ArXiv*, abs/2601.16515, 2026. URL https://api.semanticscholar.org/CorpusID:284981782.

Yifei Yu, Xiaoshan Wu, Xinting Hu, Tao Hu, Yangtian Sun, Xiaoyang Lyu, Bo Wang, Lin Ma, Yuewen Ma, Zhongrui Wang, and Xiaojuan Qi. Videossm: Autoregressive long video generation with hybrid state-space memory. *ArXiv*, abs/2512.04519, 2025. URL https://api.semanticscholar.org/CorpusID:283557042.

Mohsen Ghafoorian, Denis Korzhenkov, and Amirhossein Habibian. Attention surgery: An efficient recipe to linearize your video diffusion transformer. *ArXiv*, abs/2509.24899, 2025. URL https://api.semanticscholar.org/CorpusID:281675646.

Jintao Zhang, Haoxu Wang, Kai Jiang, Shuo Yang, Kaiwen Zheng, Haocheng Xi, Ziteng Wang, Hongzhou Zhu, Min Zhao, Ion Stoica, Joseph E. Gonzalez, Jun Zhu, and Jianfei Chen. Sla: Beyond sparsity in diffusion transformers via fine-tunable sparse-linear attention. In *International Conference on Learning Representations (ICLR)*, 2026.

Mohsen Ghafoorian and Amirhossein Habibian. Rehyat: Recurrent hybrid attention for video diffusion transformers. *ArXiv*, abs/2601.04342, 2026. URL https://api.semanticscholar.org/CorpusID:284543807.

Yanhong Li, Songlin Yang, Shawn Tan, Mayank Mishra, Rameswar Panda, Jiawei Zhou, and Yoon Kim. Distilling to hybrid attention models via kl-guided layer selection. In *International Conference on Learning Representations (ICLR)*, 2026b.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *ArXiv*, abs/2501.00103, 2024. URL https://api.semanticscholar.org/CorpusID:275212083.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, Weiming Xiong, Wei Wang, Nuo Pang, Kang Kang, Zhiheng Xu, Yuzhe Jin, Yupeng Liang, Yubing Song, Peng Zhao, Boyuan Xu, Di Qiu, Debang Li, Zhengcong Fei, Yang Li, and Yahui Zhou. Skyreels-v2: Infinite-length film generative model. *ArXiv*, abs/2504.13074, 2025. URL https://api.semanticscholar.org/CorpusID:277856899.

Wenhao Wang and Yi Yang. Vidprom: A million-scale real prompt-gallery dataset for text-to-video diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21807–21818, 2024.

Abdul Mohaimen Al Radi, Kunyang Li, Yuzhang Shang, Mubarak Shah, and Yu Tian. Keepsake: Selective spatial memory for long-horizon video generation. *arXiv preprint arXiv:2610.06588*, 2026a.

Mengxin Zheng, Jiaqi Xue, Xun Chen, YanShan Wang, Qian Lou, and Lei Jiang. Trojfsp: Trojan insertion in few-shot prompt tuning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024b.

Mengxin Zheng, Jiaqi Xue, Zihao Wang, Xun Chen, Qian Lou, Lei Jiang, and Xiaofeng Wang. Ssl-cleanse: Trojan detection and mitigation in self-supervised learning. In *European Conference on Computer Vision (ECCV)*, 2024c.

Qian Lou, Xin Liang, Jiaqi Xue, Yancheng Zhang, Rui Xie, and Mengxin Zheng. CR-UTP: Certified robustness against universal text perturbations on large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9863–9875, 2024.

Jiaqi Xue, Yifei Zhao, Mansour Al Ghanim, Shangqian Gao, Ruimin Sun, Qian Lou, and Mengxin Zheng. OpenMarkX: An extensible platform for watermarking open-source model and agent outputs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2026a. URL https://arxiv.org/abs/2510.23891.

Mansour Al Ghanim, Jiaqi Xue, Rochana Prih Hastuti, Mengxin Zheng, Yan Solihin, and Qian Lou. Evaluating the robustness and accuracy of text watermarking under real-world cross-lingual manipulations. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 7396–7416, 2025.

Yifei Zhao, Qian Lou, and Mengxin Zheng. Sif: Semantically in-distribution fingerprints for large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17399–17408, June 2026.

Jiaqi Xue, Xin Xin, Wei Zhang, Mengxin Zheng, Qianqian Song, Minxuan Zhou, Yushun Dong, Dongjie Wang, Xun Chen, Jiafeng Xie, Liqiang Wang, David Mohaisen, Hongyi Wu, and Qian Lou. SoK: Can fully homomorphic encryption support general AI computation? a functional and cost analysis. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2026b.

Jiaqi Xue, Mayank Kumar, Yuzhang Shang, Shangqian Gao, Rui Ning, Mengxin Zheng, Xiaoqian Jiang, and Qian Lou. Dictpfl: Efficient and private federated learning on encrypted gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Abdul Mohaimen Al Radi, Kunyang Li, Yuzhang Shang, Mubarak Shah, and Yu Tian. Aero-world: Action-conditioned aerial video generation from inertial controls. *arXiv preprint arXiv:2605.19728*, 2026b.

Jiaqi Xue, Qian Lou, Jiarong Xing, and Heng Huang. R2-router: A new paradigm for llm routing with reasoning. In *International Conference on Machine Learning (ICML)*, 2026c.

Jiaqi Xue, Mengxin Zheng, Heng Huang, and Qian Lou. MultiTurnRoute: A platform for building and evaluating multi-turn LLM routers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2026d. URL https://openreview.net/forum?id=CPCK7uqzh3.

Ahasan Kabir, Jiaqi Xue, Mengxin Zheng, and Qian Lou. Hw-router: Hardware-aware routing for scalable multi-llm serving. In *Design Automation Conference (DAC)*, 2026.

Jiaqi Xue, Yanjun Wang, Xiangci Li, Lingbo Mo, Aritra Sengupta, Shweta Garg, Murali Krishna Ramanathan, and Myeongsoo Kim. Aecp: Artifact-exclusive communication protocol for multi-agent code generation. *arXiv preprint arXiv:2610.06481*, 2026e.

Kunyang Li, Jeffrey A. Chan Santiago, Sarinda Dhanesh Samarasinghe, Gaowen Liu, and Mubarak Shah. Gvd: Guiding video diffusion model for scalable video distillation. *arXiv preprint arXiv:2507.22360*, 2025.

Kunyang Li, Hai Nguyen, Joshua Lowe, Chenguang Zhao, Peace C. Madueme, Mehdi Hedjazi Moghari, Mubarak Shah, Pegah Khosravi, and Yuzhang Shang. Cinemr: Tool-integrated vision-language reasoning for quantitative cardiac mri assessment. *arXiv preprint arXiv:2610.01166*, 2026c.

Ashmal Vayani, Dinura Dissanayake, Hasindri Watawana, Noor Ahsan, Nevasini Sasikumar, Omkar Thawakar, Henok Biadglign Ademtew, Yahya Hmaiti, Amandeep Kumar, Kartik Kukreja, Mykola Maslych, Wafa Al Ghallabi, Mihail Minkov Mihaylov, Chao Qin, Abdelrahman M. Shaker, Mike Zhang, Mahardika Krisna Ihsani, Amiel Gian Esplana, Monil Gokani, Shachar Mirkin, Harsh Singh, Ashay Srivastava, Endre Hamerlik, Fathinah Asma Izzati, Fadillah Adamsyah Maani, Sebastian Cavada, Jenny Chim, Rohit Gupta, Sanjay Manjunath, Kamila Zhumakhanova, Feno Heriniaina Rabevohitra, Azril Hafizi Amirudin, Muhammad Ridzuan, Daniya Najiha Abdul Kareem, Ketan Pravin More, Kunyang Li, Pramesh Shakya, Muhammad Saad, Amirpouya Ghasemaghaei, Amirbek Djanibekov, Dilshod Azizov, Branislava Jankovic, Naman Bhatia, Alvaro Cabrera, Johan Obando-Ceron, Olympiah Otieno, Febian Farestam, Muztoba Rabbani, Sanoojan Ballah, Santosh Sanjeev, Abduragim Shtanchaev, Maheen Fatima, Thao Nguyen, Amrin Kareem, Toluwani Aremu, Nathan Augusto Zacarias Xavier, Amit Bhatkal, Hawau Olamide Toyin, Aman Chadha, Hisham Cholakkal, Rao Muhammad Anwer, Michael Felsberg, Jorma Laaksonen, Thamar Solorio, Monojit Choudhury, Ivan Laptev, Mubarak Shah, Salman Khan, and Fahad Shahbaz Khan. All languages matter: Evaluating LMMs on culturally diverse 100 languages. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19565–19575, 2025.

Huili Huang, Chengeng Liu, Max Mahdi Roozbahani, and J David Frost. Seismomind: A zero-shot decision-tree based multimodal large language model framework for post-earthquake infrastructure damage understanding. *Advanced Engineering Informatics*, 77:105207, 2027.

Huili Huang, Chengeng Liu, Danrong Zhang, Shail Patel, Anastasiya Masalava, Sagar Sadak, Parisa Babolhavaeji, WeiHong Low, Max Mahdi Roozbahani, and J David Frost. Eidseg: A pixel-level semantic segmentation dataset for post-earthquake damage assessment from social media images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 38589–38597, 2026.

Huili Huang, Andrew Zhang, Danrong Zhang, Max Mahdi Roozbahani, and James David Frost. Daseg: A domain-adaptive segmentation pipeline using vision foundation models—earthquake damage detection use case. *Remote Sensing*, 17(16):2812, 2025b.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing Systems*, 37:68658–68685, 2024.

A Inference algorithm

Algorithm 1 describes our hybrid inference with a fixed-size recurrent state.

Algorithm 1 ARL^2 hybrid inference (ours)

Require: text prompt p, number of blocks N, denoising steps $T = [t_1, \ldots, t_m]$
Ensure: video frames $Y = [y_0, \ldots, y_{N-1}]$

1: $S \leftarrow \mathbf{0} \in \mathbb{R}^{H \times D \times D}$ ▷ fixed-size GDN state per hybrid layer
2: **for** $i = 0$ to $N - 1$ **do**
3: $z_i \sim \mathcal{N}(0, I)$
4: **for** $t \in T$ **do** ▷ denoising loop — state is **not** updated
5: $Q, K, V \leftarrow W_q(z_i),\ W_k(z_i),\ W_v(z_i)$
6: $O_{\text{intra}} \leftarrow \text{Softmax}(\text{RoPE}(Q),\ \text{RoPE}(K),\ V)$ ▷ intra-block only: $L \times L$
7: $Q' \leftarrow \text{L2Norm}(\text{RoPE}(\phi_q(Q)))$
8: $O_{\text{inter}} \leftarrow Q' \cdot S$ ▷ query fixed state: $O(HD^2)$
9: $O \leftarrow O_{\text{intra}} + G \odot O_{\text{inter}}$ ▷ gated combination
10: $z_i \leftarrow \text{denoise_step}(z_i, O, t)$
11: **end for**
12: $\hat{z}_i \leftarrow$ clean latent prediction ▷ clean pass — state is updated
13: $K, V \leftarrow W_k(\hat{z}_i),\ W_v(\hat{z}_i)$
14: $K' \leftarrow \text{L2Norm}(\text{RoPE}(\phi_k(K))), \quad V' \leftarrow \phi_v(V)$
15: $\alpha \leftarrow \text{gate_proj}(\hat{z}_i), \quad \beta \leftarrow \text{lr_proj}(\hat{z}_i)$ ▷ per-token forget/learning gates
16: $S \leftarrow \text{GDN_update}(S, K', V', \alpha, \beta)$ ▷ S remains $H \times D \times D$
17: $y_i \leftarrow \text{decode}(\hat{z}_i)$
18: **end for**
19: **return** Y

Algorithm 1 highlights the two key design principles of ARL^2: (1) **Frame-level access**: during each denoising step, all L tokens in the current block query the same pre-update state S, ensuring consistent history observation (lines 7–8); (2) **Clean-pass update**: the GDN state is updated only after the final clean pass (line 13), preventing noisy denoising intermediates from contaminating the persistent memory. The state S has a fixed size of $H \times D \times D$ regardless of the number of generated blocks, reducing per-layer memory from $O(N)$ to $O(1)$.

B Per-layer complexity analysis

The decomposition yields a clear complexity separation. Let b denote the block length (tokens per autoregressive block), n the number of blocks, $L_{\text{tot}} = n \cdot b$ the total sequence length, and P the number of forward passes per block (P=5 in our pipeline: 4 noisy denoising steps + 1 clean-context pass). For each forward pass through a softmax layer at block index k, the attention window spans all $k \cdot b$ cached tokens, giving per-pass FLOPs $8Bbd^2 + 4Bb^2kd$. Summing over P passes and n blocks:

$$F^{\text{video}}_{\text{softmax}} = P\sum_{k=1}^{n}\left(8Bbd^2 + 4Bb^2kd\right) = 8PBL_{\text{tot}}d^2 + 2PBbL_{\text{tot}}d(n{+}1) \ \in \ \Theta(L^2_{\text{tot}}). \tag{13}$$

For a hybrid layer, the intra-block softmax operates on a fixed $b \times b$ window regardless of k, while the inter-block query is a matrix–vector product against the $D \times D$ state. The recurrent state update (feature maps ϕ_k, ϕ_v and the FLA kernel) is executed only on the clean pass (1 of P). Per-video FLOPs:

$$F^{\text{video}}_{\text{hybrid}} = 8PBL_{\text{tot}}d^2 + 4PBbL_{\text{tot}}d + (4P{+}8)BL_{\text{tot}}dD \ \in \ \Theta(L_{\text{tot}}), \tag{14}$$

where the first term is the shared projection cost, the second is intra-block softmax (constant per block, linear in n), and the third combines the state query ($4PBL_{\text{tot}}dD$: feature map ϕ_q and $Q'S$ each pass) with the amortized state update ($8BL_{\text{tot}}dD$: feature maps ϕ_k, ϕ_v, the delta $K'S_{N-1}$, and the rank-one outer product, clean pass only). The per-block cost of the hybrid layer is independent of the block index k, which is the structural source of the quadratic-to-linear reduction.

The memory footprint follows the same pattern. Each softmax layer stores K and V caches of shape $(B, L_{\text{tot}}, H, D)$ in bfloat16, consuming $4BL_{\text{tot}}d$ bytes that grow linearly with L_{tot}. Each

hybrid layer stores only the recurrent state $S \in \mathbb{R}^{B\times H\times D\times D}$ in float32, occupying $4BHD^2$ bytes—a *constant* independent of L_{tot}. For our Wan 1.3B backbone (H=12, D=128, B=1), this amounts to 768 KiB per hybrid layer versus 192–2,304 MiB per softmax layer as n ranges from 21 to 252 blocks (a 256×–3,072× reduction).

C Evaluation details

Efficiency evaluation. Efficiency measurements (Table 2 in the main text) are conducted on a single NVIDIA RTX PRO 6000 Blackwell (96 GB). Reported generation time (`gen_s`) measures end-to-end video generation (denoising loop + VAE decode), excluding text encoding and model loading. Memory is `torch.cuda.max_memory_allocated()`, reset per video. All runs use cold-start conditions (GPU $\leq 50\,^{\circ}$C) to eliminate thermal throttling bias. Self Forcing is architecturally identical to Causal Forcing (only EMA weights differ), so their latency and memory are equal.

Stage 1 stress-test evaluation. Layer selection uses a curated set of stress-test prompts (Table 5) designed to maximally expose recurrent state weaknesses across three capability axes: temporal consistency (TC), motion dynamics (DY), and spatial quality (SQ). Each of the 30 layers is evaluated via single-layer replacement on 6 VBench quality dimensions. The resulting per-layer sensitivity scores guide which layers are safe for hybrid replacement.

Table 5: Stress-test prompts for sensitivity-guided layer selection. Each prompt targets a specific state weakness axis: TC = temporal consistency, DY = motion dynamics, SQ = spatial quality.

ID	Target Axis	Stress Category	Prompt (abbreviated)
S00	TC	Appearance retention	Woman with red striped scarf walks down cobblestone street
S01	TC	Brief occlusion	Golden retriever runs behind oak tree and re-emerges
S02	TC	Long occlusion	White SUV passes behind rocky outcropping and re-emerges
S03	TC	Background stability	Ballet dancer performs pirouette on ornate theater stage
S04	TC	Multi-entity tracking	Two puppies (golden retriever + black labrador) play on beach
S05	DY	Fast motion	Hummingbird darts between tropical flowers
S06	TC+DY	Camera orbit	Camera slowly orbits white marble statue in museum gallery
S07	SQ	Static scene	Stop motion of flower growing on windowsill
S08	SQ	Slow portrait	Victoria crowned pigeon close-up with blue plumage
S09	DY+TC	Natural dynamics	Train traveling through Tokyo suburbs with reflections
S10	DY	Group motion	Flock of paper airplanes through dense jungle canopy
S11	TC+SQ	Reflection + motion	Gray-haired man at Parisian café, cinematic lighting

D Qualitative comparison across models

Figure 7 provides a side-by-side qualitative comparison of all evaluated models on the prompt *"A 3D model of a 1800s victorian house."* We uniformly sample 6 frames (IDs 0, 16, 32, 48, 64, 80) from each 81-frame generation. LTX-Video loses structural coherence after the first few frames, collapsing into unrecognizable artifacts. Wan2.1-T2V and SANA-Video produce stylistically consistent results but exhibit limited 3D structure. SkyReels-V2 maintains temporal stability but generates a flat, low-detail scene. MAGI-1 renders a detailed Victorian house with strong temporal consistency, benefiting from its larger 4.5B parameter count. Among the 1.3B-class autoregressive models, Self Forcing and Causal Forcing both produce plausible houses. ARL^2 (50%) achieves visual quality comparable to them, with stable appearance and consistent architectural details throughout the sequence, while running $1.40\times$ faster in wall-clock time and using 31% less memory at 501 frames (Tab. 2). ARL^2 (75%) maintains coherent structure and lighting across all frames with an even larger efficiency margin: $2.26\times$ wall-clock speedup and 54% memory reduction at 921 frames, laying a strong foundation for streaming video generation. Playable video results are provided on our project page (https://lky-ang.github.io/ARL2/).

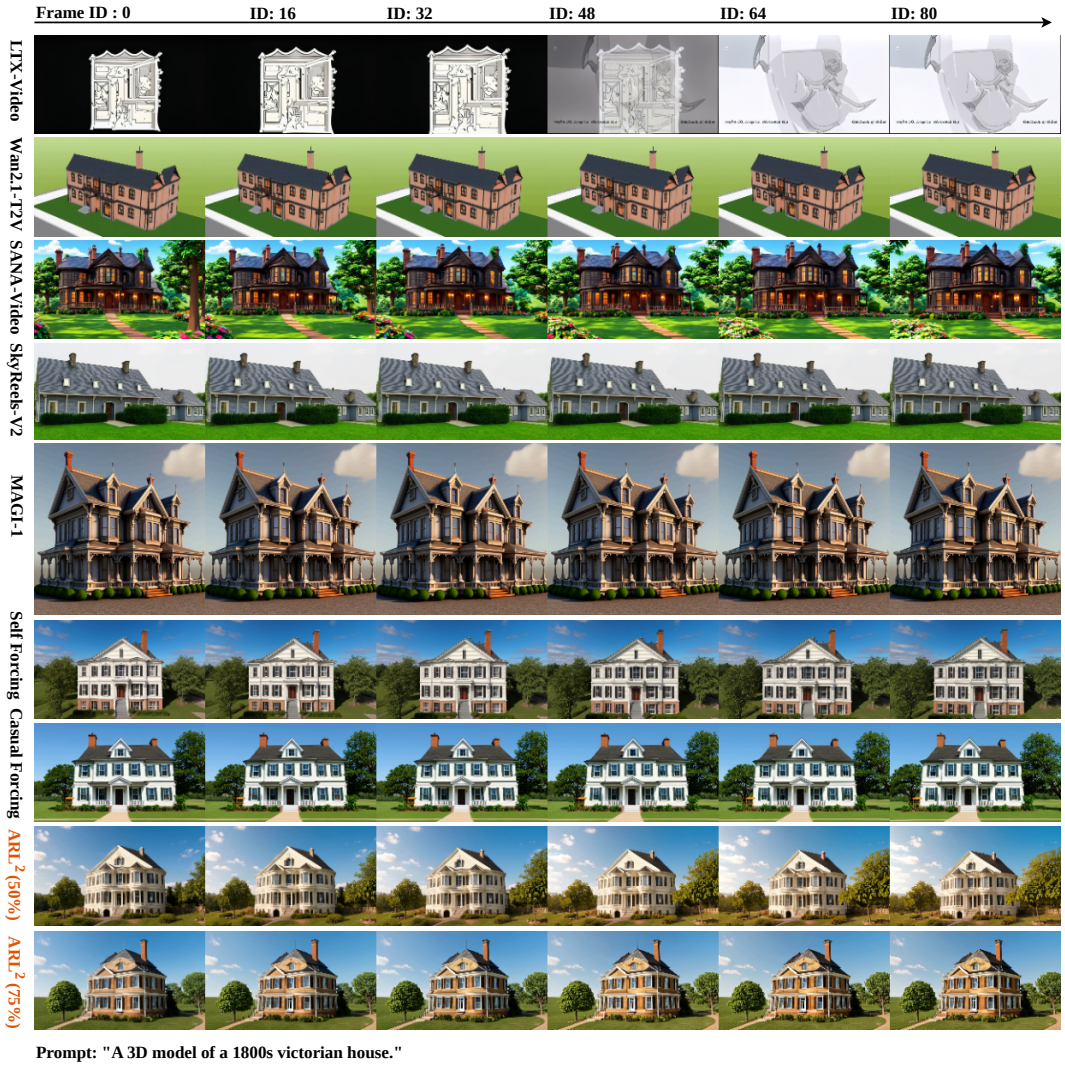


Figure 7: Qualitative comparison across all evaluated models. Six uniformly sampled frames from 81-frame videos generated with the prompt *"A 3D model of a 1800s victorian house."* ARL^2 variants maintain consistent appearance and structural detail comparable to the full-softmax baselines while operating with significantly reduced memory and compute.

E Detailed VBench per-dimension results

Table 6 reports the detailed VBench scores evaluated on the full 946-prompt official benchmark. All models generate at 832×480, 81 frames unless noted.

Table 6: Detailed VBench per-dimension scores. ‡: native resolution 544×960, 97 frames.

Dimension	SANA-Video	Wan2.1 T2V	LTX-Video	CF	SF	ARL^2 (50%)	ARL^2 (75%)	MAGI-1†	SkyReels‡
Quality dimensions									
subject_consistency	97.09	96.18	87.89	96.19	95.62	95.63	94.85	95.71	**97.78**
background_consistency	96.95	**98.33**	92.00	96.44	96.40	96.41	95.94	97.14	97.68
temporal_flickering	89.22	**98.33**	98.20	96.27	96.82	97.26	96.90	97.69	97.97
motion_smoothness	95.10	96.60	97.91	94.24	95.59	95.59	94.24	**98.67**	98.50
aesthetic_quality	**70.48**	65.35	48.08	68.80	66.75	68.29	68.44	62.44	59.66
imaging_quality	67.55	68.64	58.62	**70.62**	69.96	69.87	69.87	64.84	62.29
Semantic dimensions									
object_class	**94.86**	75.00	29.30	85.13	84.89	86.23	83.62	85.76	32.03
multiple_objects	**81.78**	64.12	8.20	64.94	68.29	67.00	68.29	50.00	15.63
human_action	92.00	62.50	62.50	79.00	80.00	77.00	72.00	**94.44**	50.00
color	88.12	85.71	97.62	90.72	93.98	91.48	91.75	86.46	**100.00**
spatial_relationship	77.46	57.92	67.88	**83.43**	78.45	78.74	78.62	76.24	38.77
scene	**73.36**	26.48	15.69	34.65	32.79	32.97	36.15	27.03	15.69
appearance_style	76.86	66.98	73.71	66.81	66.77	65.66	67.76	**87.12**	66.47
temporal_style	**67.72**	63.40	44.02	66.10	64.43	65.65	66.83	64.38	55.58
overall_consistency	74.70	64.99	62.97	67.98	66.80	66.43	67.16	**76.74**	58.00
Quality Avg	86.07	**87.24**	80.45	87.09	86.86	87.17	86.71	86.08	85.65
Semantic Avg	**80.76**	63.01	51.32	70.97	70.71	70.13	70.24	72.02	48.02
Total	**85.00**	82.39	74.62	83.87	83.63	83.77	83.41	83.27	78.12

F System Deployment

Efficiency. In practical deployment, to keep the per-step inference cost of the full model bounded regardless of video length, the softmax attention layers that are not replaced by our linear attention module can be further combined with sliding-window attention [Yang et al., 2026], sparse attention [Guo et al., 2026], KV-cache compression [Li et al., 2026a], or memory-bank management [Al Radi et al., 2026a]. As shown in Tab. 7, we adopt sliding-window and sparse attention, which reduce attention FLOPs by up to $5.83\times$ and peak GPU memory from 53.4 GB to 20.9 GB for 30 s videos while maintaining comparable quality, making long-video generation feasible on consumer-grade GPUs. Empirically, replacing 50%–65% of the layers with our linear attention module is a favorable engineering choice.

Table 7: System-level efficiency when the remaining softmax layers are also bounded. FLOPs Speedup is the theoretical reduction in attention FLOPs relative to Causal Forcing with a full KV cache. VBench score is evaluated on a subset of the VBench prompts.

Duration	Method	FLOPs Speedup↑	Peak Mem. (GB)↓	VBench Total↑
5 s	Causal Forcing, full KV (baseline)	1.00×	23.9	84.62
	Sliding window [Yang et al., 2026]	1.08×	19.4	83.64
	Sparse attention [Guo et al., 2026]	1.38×	18.9	83.51
	ARL^2 (50%) + sliding window	1.40×	18.8	**84.81**
	ARL^2 (50%) + sparse attention	**1.63×**	**18.5**	84.78
30 s	Causal Forcing, full KV (baseline)	1.00×	53.4	82.75
	Sliding window [Yang et al., 2026]	3.36×	21.8	82.25
	Sparse attention [Guo et al., 2026]	4.72×	21.3	81.15
	ARL^2 (50%) + sliding window	4.76×	21.2	**84.09**
	ARL^2 (50%) + sparse attention	**5.83×**	**20.9**	83.73

Trustworthiness. Deploying video generation models in practice also requires ensuring model integrity [Zheng et al., 2024b,c, Lou et al., 2024], protecting intellectual property and content provenance [Xue et al., 2026a, Al Ghanim et al., 2025, Zhao et al., 2026], and preserving data privacy [Xue et al., 2026b, 2025].

Applications. Our efficiency technique can be applied to a range of downstream tasks, such as interactive video generation [Al Radi et al., 2026b], LLM routing [Xue et al., 2026c,d, Kabir et al., 2026], multi-agent systems [Xue et al., 2026e], video dataset distillation [Li et al., 2025], multimodal understanding across medical and multilingual domains [Li et al., 2026c, Vayani et al., 2025], and disaster detection with foundation models [Huang et al., 2027, 2026, 2025b].

G Broader impact and limitations

Broader impact. ARL^2 reduces the computational and memory cost of autoregressive video generation by up to $2.26\times$ and 54%, respectively. This efficiency gain lowers the hardware barrier for long-form video synthesis, potentially democratizing access to high-quality video generation for researchers and practitioners with limited GPU resources. By enabling longer videos on a single consumer-grade GPU, the method may accelerate applications in education, content creation, and accessibility tools. However, as with all generative video models, the technology could be misused to produce misleading or harmful synthetic media. We encourage the community to develop and deploy appropriate watermarking, provenance tracking, and content authentication mechanisms alongside efficient generation methods.

Limitations. The recurrent state $S \in \mathbb{R}^{H\times D\times D}$ has a fixed capacity regardless of video length. While this constant footprint is the source of the $O(1)$ memory advantage, it may limit the model's ability to retain fine-grained details over very long horizons involving hundreds of distinct scenes. In addition, kernel support for linear attention is less mature than that for softmax attention, which benefits from highly optimized implementations such as FlashAttention [Shah et al., 2024]. As a result, the wall-clock speedup on short videos is limited. Developing more efficient linear attention kernels remains an important direction for future work, and we hope this work motivates broader community efforts in that direction.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction claim constant-memory inference, up to $2.26\times$ speedup and 54% memory reduction, and quality matching the softmax teacher. These are validated experimentally in Sec. 4 (Tables 1–2) and analytically in App. B.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: App. G provides a dedicated discussion of limitations, including the fixed capacity of the recurrent state over very long horizons and the less mature kernel optimization for linear attention.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The per-layer complexity analysis (App. B) provides a complete derivation of FLOPs and memory for both softmax and hybrid layers, with all variables defined and assumptions (batch size, number of passes, block length) stated explicitly.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The hybrid module architecture is fully specified in Sec. 3, and the training setup (data, training steps, and compute) is described in Sec. 4.1. The evaluation protocol is detailed in Sec. 4.1 and App. C. Full configurations and implementation details are available in our public code repository (https://github.com/lky-ang/ARL2).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our code and pretrained checkpoints are publicly available at https://github.com/lky-ang/ARL2 and https://huggingface.co/lky-ang/ARL2, with instructions for installation and inference.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details including data sources, training steps, and compute are provided in Sec. 4.1, and full configurations are available in our public code repository (https://github.com/lky-ang/ARL2). Evaluation details including the VBench protocol, prompt subsets, and measurement methodology are in Sec. 4.1 and App. C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: VBench evaluations use a single seed (seed 0) with 1 sample per prompt over 946 prompts, following the standard VBench protocol. While we do not report multi-seed variance due to computational constraints, VBench is evaluated over 946 prompts, which provides a large sample size and yields stable aggregate metrics.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Sec. 4.1 reports the total training compute ($\sim$156 H100 GPU-hours, $\sim$50 for Stage 1 and $\sim$106 for Stage 2). Efficiency evaluation hardware (RTX PRO 6000 Blackwell 96 GB) is stated in Sec. 4.1 and App. C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research focuses on improving efficiency of video generation models. No human subjects are involved. Training uses only publicly available text prompts from VidProM and videos synthesized by the teacher model; no external video datasets are used.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: App. G discusses both positive impacts (lowering the hardware barrier for long-form video synthesis, enabling applications in education and content creation) and potential negative impacts (misuse for synthetic media), along with recommended mitigation strategies such as watermarking and provenance tracking.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released checkpoints are efficiency-oriented variants distilled from the publicly available Wan2.1-T2V-1.3B model and do not add generation capabilities beyond it; no new datasets are released. The release therefore poses no additional misuse risk beyond the existing public base model.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The base model (Wan2.1-T2V-1.3B), codebase (Causal Forcing), evaluation benchmark (VBench), training data source (VidProM), and all baseline methods are properly cited. Wan2.1, Causal Forcing, and VBench are released under Apache 2.0; VidProM is released under CC BY-NC 4.0 and is used here for non-commercial research only.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release our code and the ARL^2 checkpoints (50% and 75% hybrid) at `https://github.com/lky-ang/ARL2` and `https://huggingface.co/lky-ang/ARL2` under the Apache 2.0 license. The repository README documents installation, checkpoint download, and inference usage. No new datasets are released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or human subjects research.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a component of the core method. Any LLM usage was limited to writing assistance and does not impact the methodology or results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.