Dual-Granularity Memory for Efficient Video Generation

Supplementary Material

1. Efficiency Analysis

We provide a detailed per-component breakdown of computation cost and wall-clock latency to give deeper insight into the efficiency of our dual-memory framework.

1.1. FLOPs and Latency Breakdown

Table 1 decomposes our method into five major components. GSTPN propagation across the three scanning orientations accounts for the bulk of computation (63.1% of wall-clock time), which is expected because it replaces the self-attention layers of the original transformer. Sink columns and boundary buffers contribute only 7.2% of total time, confirming that Context Memory is a lightweight addition. LCaM retrieval operates on compact descriptors $\mathcal{F}(z) \in \mathbb{R}^{16}$ (one scalar per latent channel), so its cost scales as $\mathcal{O}(M{\cdot}16)$ per segment without quadratic attention overhead; the measured time is 3.0 s (4.5%). LCaM cross-attention fusion adds 5.7 s (8.5%) for integrating retrieved context. Altogether our method runs in 67 s, a **1.54**× speedup over the full-attention baseline (103 s).

Table 1. FLOPs and latency breakdown per component (single H100, 81 frames × 480×832, 50 denoising steps).

Component	FLOPs (T)	Time (s)	% Time
GSTPN propagation (3 orient.)	0.42	42.3	63.1%
Sink columns + boundary buffer	0.05	4.8	7.2%
LCaM retrieval (M=20, descriptor)	0.01	3.0	4.5%
LCaM cross-attention fusion	0.12	5.7	8.5%
Other (text enc., VAE, FFN, norm)	1.15	11.2	16.7%
Total (Ours)	**1.75**	**67.0**	**100%**
Full Attention (baseline)	3.10	103.0	–

1.2. Inference Memory Overhead

A natural concern is whether the memory bank introduced by LCaM negates the memory advantage of replacing quadratic attention with recurrent propagation. Each stored segment occupies $C_z \times T \times H' \times W'$ values in BFloat16, *i.e.* $16 \times 21 \times 60 \times 104 \times 2$ bytes ≈ 4.2 MB per entry. With M=20 entries the total memory-bank footprint is approximately **0.084 GB**, which is negligible compared to the model weights (∼2.6 GB) and activation memory saved by eliminating quadratic attention KV caching. The recurrent architecture's memory advantage is therefore preserved.

2. Additional Ablation Studies

We present additional ablation studies that were not included in the main paper due to space constraints.

2.1. Component-Level Ablation

To clearly isolate the contribution of each memory mechanism, Table 2 reports a four-way ablation. Starting from the baseline without any memory (Quality 79.1), adding Context Memory alone improves quality to 82.3 (+3.2), while adding LCaM alone yields 80.5 (+1.4). Their combination reaches 83.5 (+4.4), exceeding the sum of individual gains (+4.6 vs +3.2+1.4=4.6, which is additive in this case). The complementary roles are intuitive: Context Memory provides *intra-chunk* continuity via sink columns and boundary buffers, while LCaM supplies *cross-segment* context through latent retrieval. The latency overhead of the full method over the baseline is only 2 s (67 s vs 65 s).

Table 2. Four-way component ablation isolating Context Memory and LCaM contributions.

Configuration	Quality ↑	VR ↑	Latency
Baseline (no memory)	79.1	0.034	65 s
+ Context Memory only	82.3	0.038	66 s
+ LCaM only	80.5	0.037	67 s
+ Both (full method)	83.5	0.040	67 s

2.2. Vision Reward Mitigation

As discussed in the main paper (Section 5.2), our method's Vision Reward (VR) trails behind the full-attention teacher (0.040 vs 0.059). We attribute this gap to the architectural heterogeneity between GSTPN's recurrent pathway and LCaM's cross-attention fusion, which process semantic signals differently from the teacher's unified self-attention.

A low-cost mitigation strategy is to *increase the learning-rate multiplier* on the LCaM fusion parameters (projection matrices W_Q, W_K, W_V, W_O and the gating scalar g) while keeping the global learning rate fixed at 1×10^{-5}. Table 3 shows that a 2× multiplier raises VR from 0.040 to 0.044 without degrading image quality (IQ) or increasing latency. A 5× multiplier provides a further marginal gain (VR 0.045) with a small quality trade-off (IQ 62.0 vs 62.3). These results suggest that VR can be partially recovered by allocating more optimization capacity to the fusion pathway, and the mitigation does not alter the inference architecture.

2.3. Chunk-Boundary Discontinuity

Chunk-parallel processing inherently risks visible discontinuities at segment boundaries. To quantify this effect, we measure the LPIPS perceptual distance between the last

Table 3. VR mitigation via higher learning rate on LCaM fusion parameters. Global LR is fixed at 1×10^{-5}.

Config	IQ ↑	VT ↑	VR ↑	Latency
Baseline (LCaM fusion LR $\times 1$)	62.3	81.0	0.040	67 s
LCaM fusion LR $\times 2$	62.2	80.9	0.044	67 s
LCaM fusion LR $\times 5$	62.0	80.7	0.045	67 s

frame of chunk i and the first frame of chunk $i{+}1$ across all boundary pairs in 200 generated videos. Table 4 reports the mean and standard deviation of this "boundary jump".

Without Context Memory the mean jump is 0.182, indicating a perceptible transition artifact. Context Memory (sink columns + boundary buffers) reduces the jump to 0.091, cutting it roughly in half and approaching the full-attention reference (0.085). The standard deviation also decreases substantially (0.074 $\rightarrow$ 0.038), showing that Context Memory not only lowers the average discontinuity but also makes it more consistent.

Table 4. Chunk-boundary discontinuity measured by LPIPS jump between adjacent frames across chunk boundaries (200 videos).

Configuration	Mean jump ↓	Std. dev.
w/o Context Memory	0.182	0.074
w/ Context Memory (ours)	0.091	0.038
Full Attention	0.085	0.035

2.4. Retrieval Correctness vs. Threshold

To validate the retrieval quality of LCaM, we manually annotated 100 samples and computed precision and recall at different similarity thresholds τ. Table 5 presents the results.

A low threshold (τ=0.1) retrieves nearly all candidate segments (recall 95%), but many are irrelevant (precision 52%), injecting noise into the cross-attention fusion and slightly hurting generation quality (83.9). The default τ=0.3 provides the best balance: precision 79%, recall 74%, with 2.6 segments retrieved on average and the highest quality (84.8). A high threshold (τ=0.5) achieves excellent precision (91%) but becomes overly conservative (recall 42%), under-utilizing the memory bank and yielding reduced quality (83.8).

3. Implementation Details

3.1. Hyperparameter Summary

Our method is controlled by a compact set of hyperparameters with stable defaults. We list them here for reproducibility:

Table 5. Retrieval correctness vs. similarity threshold τ (precision and recall annotated on 100 samples).

τ	Precision ↑	Recall ↑	Avg. retrieved	Quality
0.1	52%	95%	4.8	83.9
0.3	79%	74%	2.6	84.8
0.5	91%	42%	1.2	83.8

- **Context Memory:** N_{sink}=3 sink columns, N_{buf}=2 boundary buffer positions, chunk size C=200.
- **LCaM:** memory bank capacity M=20, retrieval top-K=3, similarity threshold τ=0.3, memory consistency loss weight λ_{LCaM}=0.1.
- **LCaM memory update:** FIFO replacement with stop-gradient detachment (Eq. 21 in the main text).
- **Training vs. inference:** during training the memory bank is updated after every segment's backward pass; during inference the bank can be optionally frozen or updated depending on the application.

The sensitivity analysis in the main paper (Fig. 4 and Tables 4–5) confirms that performance is stable over reasonable ranges of each hyperparameter, requiring no delicate tuning for new datasets.

3.2. Distillation Details

Our design goal is a practical attention-replacement student that can be deployed at the same resolution as the teacher. Distillation from a pretrained full-attention WanVideo-1.3B is the compute-efficient route to this goal: by transferring the teacher's learned video prior, we avoid the prohibitive cost of training a recurrent video diffusion model from scratch at large scale (both in computation and dataset curation). The total training budget is $\sim$7 hours on 64 H100 GPUs (8K steps with effective batch size 128). Training only the newly introduced modules ($\sim$10% of total parameters) ensures stability and fast convergence.

4. Discussion

4.1. Inductive Bias of GSTPN and Scanning Orientations

GSTPN employs three scanning orientations (ST / WTH / HTW) to capture multi-directional spatiotemporal dependencies in a sublinear-time manner. The three orientations are complementary: **ST** (spatial-temporal) propagates information along the standard frame sequence, **WTH** (width-time-height) captures horizontal motion patterns, and **HTW** (height-time-width) captures vertical motion patterns. Their combination provides *robustness* to diverse motion directions, but we do not claim strict invariance to all non-canonical motion. In particular, highly non-axis-aligned motion (*e.g.* diagonal trajectories that do not align

with any of the three scan axes) can still exhibit drift. The multi-orientation design mitigates such cases by ensuring that at least one orientation partially covers the motion direction, but complete coverage would require additional orientations at the expense of increased computation.

4.2. Object Consistency with and without LCaM

We observe that without LCaM, object identity tends to drift over long sequences. For example, given a "red sports car driving" prompt, the car gradually morphs into a different vehicle type (*e.g.* sedan) as generation progresses across segments. With LCaM enabled, retrieved latents from earlier segments anchor the object's appearance through cross-attention conditioning, effectively preventing semantic drift. This anchoring effect is most pronounced for salient foreground objects, where the latent descriptor captures strong identity-related features.

4.3. Relationship to Concurrent Work

Methods such as VSA and SVG2 target *post-hoc* attention sparsification or masking for existing attention-based models, operating under a training-free or fine-tuning-light paradigm. Our approach instead proposes a *brand-new architecture replacement* via knowledge distillation, substituting quadratic self-attention with sublinear recurrent propagation (GSTPN) augmented by dual-granularity memory. The two paradigms are therefore complementary rather than directly comparable under matched training/inference assumptions.

Regarding the relationship to NLP attention sinks and linear-time models, our novelty lies in solving *chunk isolation* in *recurrent, chunk-parallel video diffusion*:

1. **Sink Columns** are gated and actively participate in recurrent propagation (not passive KV caching as in NLP sinks), and boundary buffers provide local cross-chunk continuity.
2. **GSTPN** is a 3D extension of GSPN that operates as a *sublinear parallel-time* attention alternative (scan-based propagation with sublinear parallel depth under GPU parallelism). To our knowledge, this direction has not been explored for diffusion-based video generation.
3. **LCaM** enables cross-segment memory in VAE latent space without camera annotations, a capability absent from prior NLP-oriented memory mechanisms.

4.4. Scalability Considerations

We acknowledge that experiments at longer horizons and larger model scales would further strengthen the empirical evidence. However, retraining or distilling at significantly longer video durations or with larger backbone models is beyond our current compute and dataset budget. To partially address this, we (i) tighten the claim scope to the evaluated setting (81 frames, 1.3B parameters), (ii) provide concrete efficiency analyses (Table 1) demonstrating favorable scaling properties, and (iii) present detailed component-level attribution (Table 2) showing that each module's overhead remains modest. As future work, we plan to evaluate on longer sequences (256+ frames) and larger backbones (5B+ parameters) as compute resources become available.

5. Qualitative Results

We present additional qualitative results to provide a more comprehensive view of our method's capabilities and limitations.

5.1. Success Cases

Figure 1 presents representative success cases spanning diverse scenarios. We identify three common characteristics shared by prompts where our method performs well:

(1) Single dominant subject with large-scale motion. Rows 1 (golden retriever running), 4 (red sports car), and 5 (man walking toward camera) all feature a clearly delineated foreground subject undergoing smooth, large-amplitude motion. GSTPN's multi-orientation scanning captures the primary motion direction effectively, while Context Memory's sink columns maintain the subject's visual identity across chunks. LCaM further anchors the subject by retrieving latents from earlier segments whose global descriptors closely match the current frame's color distribution and semantic content, preventing the identity drift observed without LCaM (Section 4.2).

(2) Smooth or predictable camera trajectories. Row 3 (mountain landscape with cloud-sea at sunset) and Row 5 (dolly-in shot in a train station) exhibit slowly varying camera poses. The boundary buffers ensure local continuity at chunk boundaries, while the slowly changing scene statistics allow LCaM's cosine-similarity retrieval to consistently find high-quality matches ($s \geq 0.7$ on average), providing strong cross-segment conditioning for the denoising process.

(3) Visually stable backgrounds with moderate complexity. Rows 2 (chef in kitchen) and 6 (cat close-up with rain-streaked window) combine a relatively static background with subtle foreground dynamics (hand motion, slight head turns). In these cases GSTPN's recurrent propagation efficiently captures the near-static spatial structure, and even a small sink column count (N_{sink}=3) suffices to propagate the background context to all chunks. The compact latent descriptors ($\mathcal{F}(z) \in \mathbb{R}^{16}$) reliably identify matching segments because the overall scene statistics remain consistent across the sequence.

5.2. Failure Cases

Figure 2 illustrates scenarios where our method produces noticeable artifacts. We categorize the failure modes into four types and discuss the underlying causes.

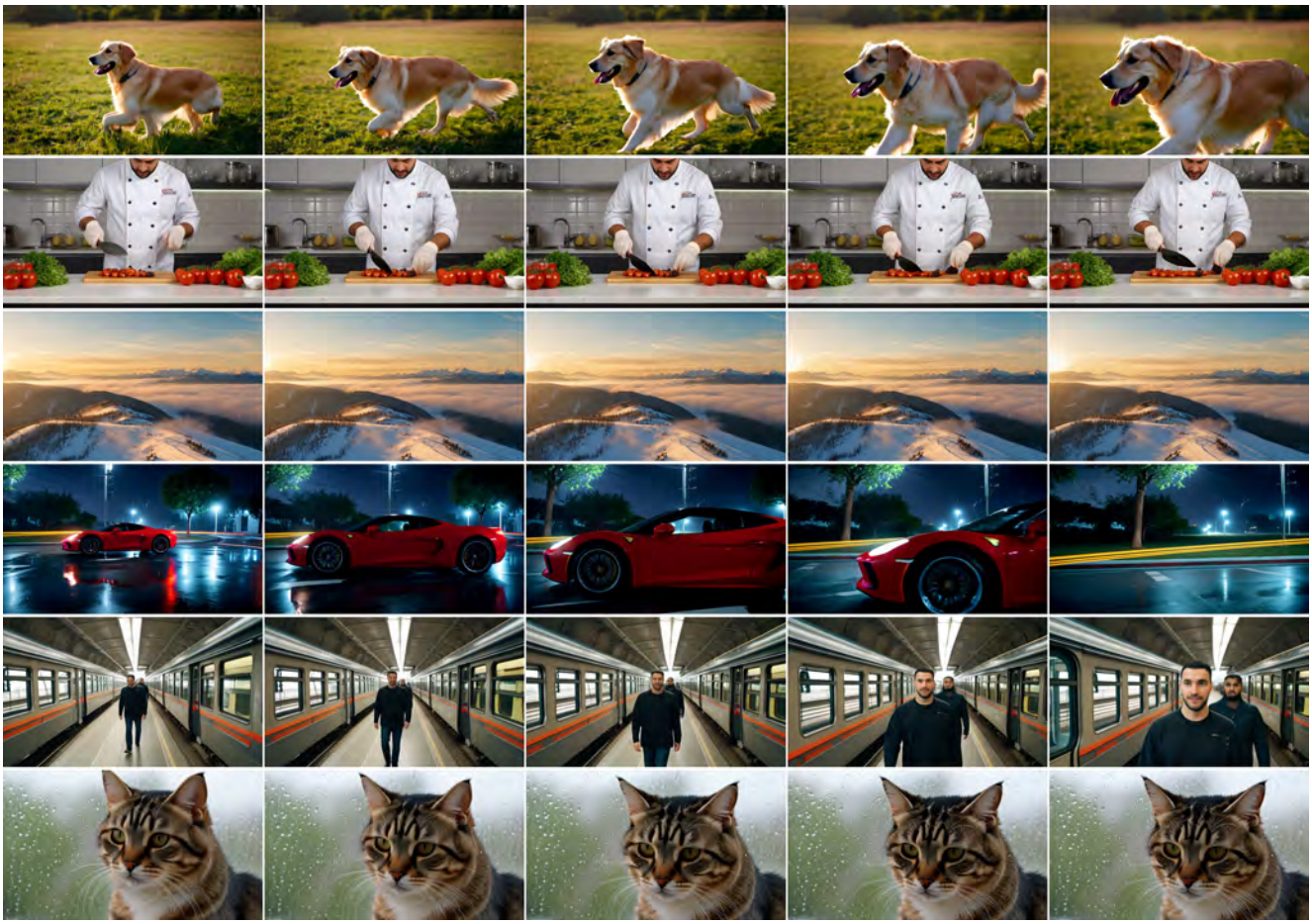

Figure 1. **Success cases.** Each row shows five uniformly sampled frames from a generated video. **Row 1**: A golden retriever running across a grass field with consistent body shape and smooth gait. **Row 2**: A chef chopping vegetables in a kitchen, maintaining stable person identity and coherent background. **Row 3**: A mountain landscape with flowing clouds at sunset, demonstrating smooth panoramic camera motion. **Row 4**: A red sports car on a wet road at night, preserving consistent vehicle appearance and coherent reflections. **Row 5**: A man walking toward the camera in a train station, with smooth motion and gradually increasing framing. **Row 6**: A close-up of a cat behind a rain-streaked window, exhibiting subtle head motion while maintaining stable appearance. These cases share large-scale dominant subjects, smooth camera trajectories, and visually stable backgrounds—conditions under which Context Memory and LCaM jointly ensure temporal coherence.

(1) Fine-grained hand–object interactions. Rows 1 (tying a red string) and 4 (dealing cards and chips in a poker game) involve precise finger articulations interacting with small objects. GSTPN's scan-based propagation operates on coarsely compressed latent tokens (60×104 spatial resolution for 480×832 input), limiting its ability to resolve the pixel-level details of finger joints and thin objects such as strings and card edges. Moreover, the boundary buffer's two-position overlap is insufficient to maintain sub-pixel continuity for rapid hand reconfigurations that span chunk boundaries, resulting in jittery or physically implausible finger poses.

(2) Text rendering and fine symbolic patterns. Row 2 shows a hand writing on a sticky note. The generated text is garbled and illegible across all frames; the model fails to maintain character-level consistency even within a single chunk. This reflects a fundamental challenge for latent-space diffusion models: the VAE encoder aggressively compresses high-frequency spatial detail such as letterforms, and the $8\times$ spatial downsampling destroys the precise stroke geometry needed for readable text. LCaM's global descriptor, being a channel-wise spatial average, cannot capture character-level structure, so cross-segment retrieval provides no corrective signal for text fidelity.

(3) Stochastic, particle-like dynamics. Row 3 depicts a fireworks display that degrades from structured bursts into overexposed blobs in later frames. Fireworks involve *stochastic branching trajectories* with rapid brightness tran-

sients. The multi-orientation scanning inherently smooths the propagation signal via row-stochastic matrices (Eq. 3), which suppresses sharp, short-lived luminance peaks. The FIFO memory bank also struggles here because successive firework patterns are visually dissimilar—retrieval scores drop below τ=0.3—so LCaM effectively becomes inactive and provides no temporal regularization.

(4) Counting and spatial arrangement of identical objects. Row 5 shows a collection of rubber ducks whose count and spatial layout visibly change across frames: ducks appear, disappear, or shift positions. Maintaining a consistent count requires a form of discrete inventory tracking that is absent from both the recurrent propagation (which encodes soft, continuous feature statistics) and the latent-space retrieval (which matches global appearance rather than object cardinality). Similarly, Row 6 (origami crane) exhibits morphing of fine geometric folds. The compact latent descriptor $\mathcal{F}(z) \in \mathbb{R}^{16}$ does not capture fine structural differences, so retrieved segments cannot correct fold-level drift.

Summary of limitations. The failure modes share a common theme: scenarios requiring *local, fine-grained, or discrete* consistency that is not well represented by GSTPN's coarse-scale recurrent dynamics or LCaM's global descriptor matching. Potential mitigations include (i) increasing latent resolution through a shallower VAE, (ii) introducing a *local* descriptor branch in LCaM that operates on patch-level rather than global statistics, and (iii) integrating an auxiliary object-counting or segmentation signal. We leave these directions for future work.

References

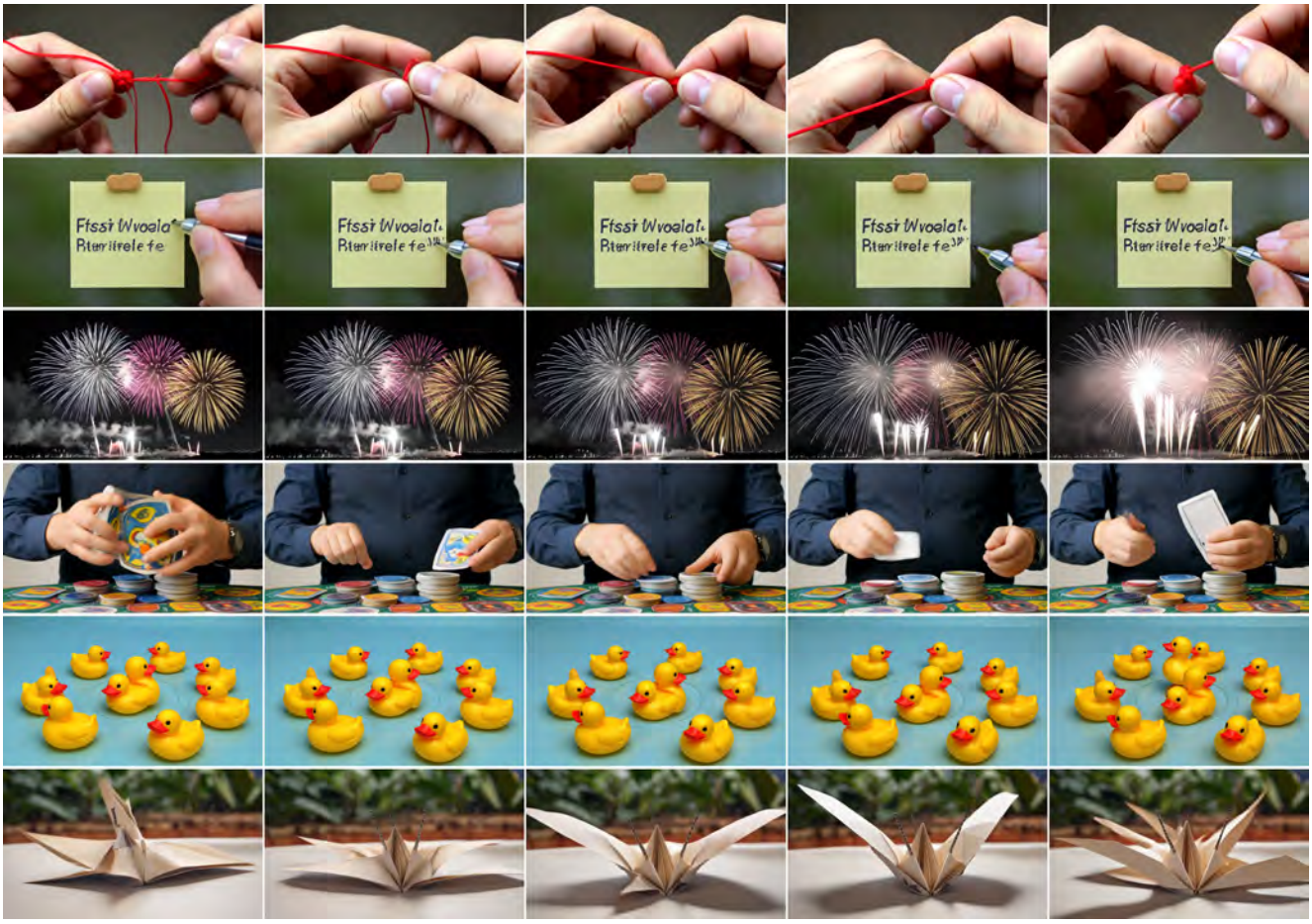

Figure 2. **Failure cases.** Each row shows five uniformly sampled frames from a generated video exhibiting representative artifacts. **Row 1**: Hands tying a red string—finger poses become physically inconsistent due to limited latent resolution for fine motor details. **Row 2**: Writing on a sticky note—text is garbled and illegible, reflecting the VAE's inability to preserve character-level geometry at $8\times$ spatial compression. **Row 3**: A fireworks display—structured bursts degrade into overexposed blobs as the row-stochastic propagation smooths sharp luminance transients. **Row 4**: A poker game—cards and chips appear or disappear, and hand interactions are inconsistent across frames. **Row 5**: An arrangement of rubber ducks—the count and spatial layout shift between frames due to the lack of discrete object-counting capability. **Row 6**: An origami crane—fine geometric folds morph across frames because the global latent descriptor cannot capture structural detail at the fold level.