MOSAICHUNK: COMPOSITING SPATIO-TEMPORAL MEMORY FOR AUTOREGRESSIVE VIDEO GENERATION

Anonymous authors
Paper under double-blind review

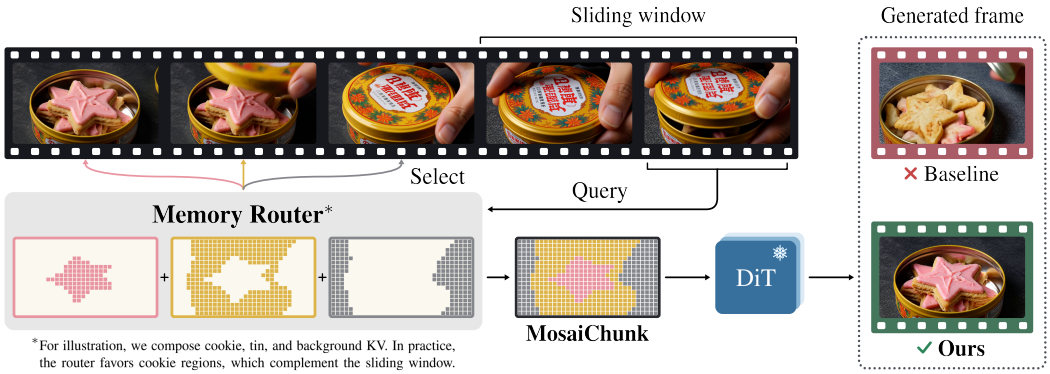

Figure 1: In text-to-video generation, prompts like *"Close the tin, then reopen it."* illustrate a well-known problem: once a frame (e.g. tin with cookie) is dropped from the context sliding window, its contents are forgotten and must be hallucinated (top). We introduce *MosaiChunk* (bottom), a spatio-temporal memory mechanism that conditions a video generation model on composed historical KV sections (pink for the cookie, yellow for the tin, and gray for the background). Our mechanism preserves fine-grained visual details over long-horizon rollouts, such as the star cookie's appearance when the tin closes and reopens.

ABSTRACT

Long-horizon autoregressive video generation is limited by a finite context window. When an object or scene falls out of context, its fine-grained visual details may be lost and difficult to recover upon reappearance. To retain access to such visual details, we introduce *MosaiChunk*, a spatio-temporal memory mechanism that composes a mosaic of selected historical key–value (KV) entries across space and time. Our approach is motivated by the observation that a frozen video generator can directly consume such non-contiguous historical KV and recover the corresponding visual content. We therefore keep the generator fixed and learn only a lightweight router that determines which historical sections to include in the mosaic under a fixed active-memory budget. We further introduce *RememBench*, a benchmark of long-horizon revisits with prompt-driven text-to-video (T2V) and camera-driven image-to-video (I2V) splits. Our experiments show that *MosaiChunk* consistently improves revisit consistency over both sliding-window inference and whole-chunk retrieval under matched memory budgets, across both T2V and I2V settings. All code and data will be released.

1 INTRODUCTION

> **"Nothing comes from nothing."**
> —*The Sound of Music* (1965)

The visual world is full of recurrence: objects return, surfaces reappear from new viewpoints, and details established at one moment become relevant again much later. For decades, vision and graphics methods have exploited this structure by constructing new imagery from visual elements that have already been observed—reusing patches, frames, and regions rather than synthesizing every

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

detail from scratch (Schödl et al., 2000; Efros & Freeman, 2001; Jojic et al., 2003; Isola & Liu, 2013). In this work, we ask: can the same principle of visual reuse support persistent memory in autoregressive video generation?

Consider the example illustrated in Figure 1, where a video model is prompted to show a biscuit tin, close it, and reopen it a few seconds later. When the tin reopens, the model should be able to reuse the biscuit's previously generated appearance, since it has already established what is inside. Yet long-horizon autoregressive video generation typically relies on sliding-window inference, which keeps the active cache bounded by retaining only the most recent entries and evicting older ones. Once the biscuit leaves this window, the generator loses direct access to that specific appearance, and may synthesize a different biscuit when the tin reopens.

The simplest solution would be to keep the entire generated history active, but this quickly becomes impractical: as the rollout grows, so do both the KV cache and the cost of attending to it. The challenge is therefore not to remember everything, but to recover a small set of historical KV needed for the current continuation. Recent approaches address this problem by retrieving the most relevant historical chunks, typically according to similarity with the current context (Cai et al., 2025; Yi et al., 2026). However, chunks bundle the keys and values of multiple frames, while the reusable visual content may occupy only a small subset of entries within these chunks.

Accordingly, we introduce *MosaiChunk*, which constructs a new kind of memory "chunk" by assembling a mosaic of selected historical keys and values that come from different spatial regions and different moments in the generated history. For example, historical KV sections corresponding to the biscuit are composited with more recent sections corresponding to the tin and background in Figure 1. Our memory mechanism is enabled by the observation that a frozen video generator can already consume such non-contiguous historical KV and use it to recover the corresponding visual content. This allows us to keep the generator fixed and train only a lightweight router to retrieve and compose the historical sections most relevant to each continuation, thereby preserving previously established visual details under a fixed active-memory budget.

Because temporally neighboring sections can have similar keys, direct similarity-based retrieval can favor recent content and miss older content needed when previously seen objects or scenes reappear. To reduce this recency bias, we train the router's descriptor encoder through self-distillation, where both teacher and student use the same frozen backbone and noisy latent, differing only in their far memory. The teacher receives oracle memory containing whole departure chunks with the content to be redrawn; the student receives a smaller memory assembled by the router. Matching their denoising velocity predictions trains the descriptors that determine section selection and weighting. Because the targets come from the backbone's own predictions on rollout latents, this oracle-memory distillation requires no rendered or recorded ground-truth target videos.

To evaluate *MosaiChunk*, we introduce *RememBench*, a benchmark of long-horizon *revisits*, in which previously seen content comes back into view. Such revisits provide a concrete test of a video model's memory: the returning content should preserve the details established earlier in the rollout. Autoregressive video models such as RAVEN-adapted MiniMax-H3 (H3-AR) (MiniMax, 2026; Lu et al., 2026) and LingBot-World-Infinity (Gao et al., 2026) support long interactive rollouts through prompt updates and camera controls. We leverage these interfaces to construct complementary prompt-driven T2V and camera-driven I2V splits. Compared with an existing whole-chunk retrieval and conditioning baseline, *MosaiChunk* improves median revisit CLIP similarity from 0.841 to 0.936 on T2V and from 0.807 to 0.863 on I2V after the camera turns 180° away and then back.

In summary, our contributions include: (i) a compositional visual memory mechanism that retrieves historical KV sections under a fixed active budget; (ii) a self-distillation strategy for learning KV section descriptions without rendered ground-truth target videos; and (iii) *RememBench*, a benchmark of long-horizon revisits with prompt-driven T2V and camera-driven I2V splits.

2 Related Work

Autoregressive video generation. Recent work has developed autoregressive video diffusion models for streaming and long-horizon generation. Diffusion Forcing (Chen et al., 2024) combines causal prediction with independently noised tokens, while CausVid (Yin et al., 2025) distills a bidirectional teacher into a few-step causal generator. Subsequent methods improve rollout training:

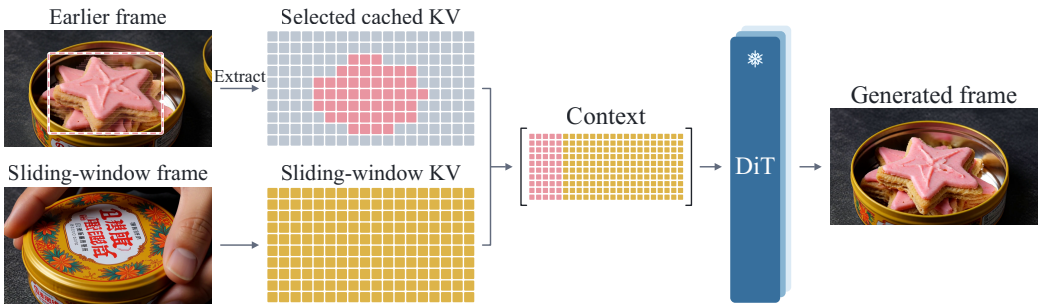

Figure 2: **Can a frozen backbone reuse a non-contiguous subset of historical KV?** To test whether a frozen video generator can consume a non-contiguous subset of KV entries, we select the KV at positions corresponding to the marked cookie region. Only the colored entries are supplied alongside the sliding window. The frozen video generator preserves the cookie's appearance when the tin opens again.

Self Forcing (Huang et al., 2025) uses self-generated histories, MV-Forcing (Fiebelman et al., 2026) extends self-forcing to joint temporal and view-wise autoregression, RAVEN (Lu et al., 2026) allows later-chunk losses to supervise historical representations, and LongLive (Yang et al., 2025) trains on longer rollouts with prompt transitions. Sliding-window inference nevertheless evicts historical KV needed for revisits. We address this through learned retrieval under a fixed active cache budget.

Long-term visual memory. Prior methods extend visual memory through frame retrieval or geometric organization of past observations (Yu et al., 2025; Xiao et al., 2025; Wu et al., 2025a), or through compressed and learned memory representations (Zhang et al., 2025; Hong et al., 2025; Wu et al., 2025b). Within native KV representations, cache-management methods reduce storage and attention costs through quantization, token selection, merging, and compaction (Xi et al., 2026; Chen et al., 2026; Luo et al., 2026; Ji et al., 2026; Li et al., 2026; Yi et al., 2025).

Retrieval-based methods select historical context using content similarity, camera/action correspondence, or learned selection mechanisms, retrieving frames, chunks, or blocks to condition subsequent generation (Cai et al., 2025; Yi et al., 2026; Wu et al., 2026; Zhao et al., 2026). Training these memory mechanisms can involve jointly optimizing memory selection and the generator, or fine-tuning the generator on revisit sequences (Zhao et al., 2026; Xue et al., 2026; Chen et al., 2026). In contrast, *MosaiChunk* retrieves fine-grained, content-based sections of original KV under a fixed active cache budget. We keep the backbone frozen and train only the selector using denoising predictions under oracle context, without rendered ground-truth target videos.

Representing visual content with reusable components. Earlier work synthesizes images and videos by reusing visual fragments, from texture patches and video frames (Efros & Freeman, 2001; Schödl et al., 2000) to regions retrieved for completion and scene composition (Wexler et al., 2004; Hays & Efros, 2007; Isola & Liu, 2013). Related approaches capture recurring appearance through epitomic representations (Jojic et al., 2003), discover objects from local-feature statistics (Sivic et al., 2005), or measure similarity through composition (Boiman & Irani, 2006). We adopt this perspective of decomposing visual content into reusable components, organizing historical KV into content-based sections that can be retrieved and composed across time.

3 Method

We first examine how a frozen video backbone can use non-contiguous historical KV entries (Section 3.1). This observation motivates *MosaiChunk*'s memory architecture and the training strategy for its descriptor encoder (Section 3.2).

3.1 Reusing Non-Contiguous Historical KV for Visual Memory

Autoregressive video models generate one chunk at a time, attending to cached keys and values (KV) from earlier chunks. Under sliding-window inference, the backbone is conditioned on the

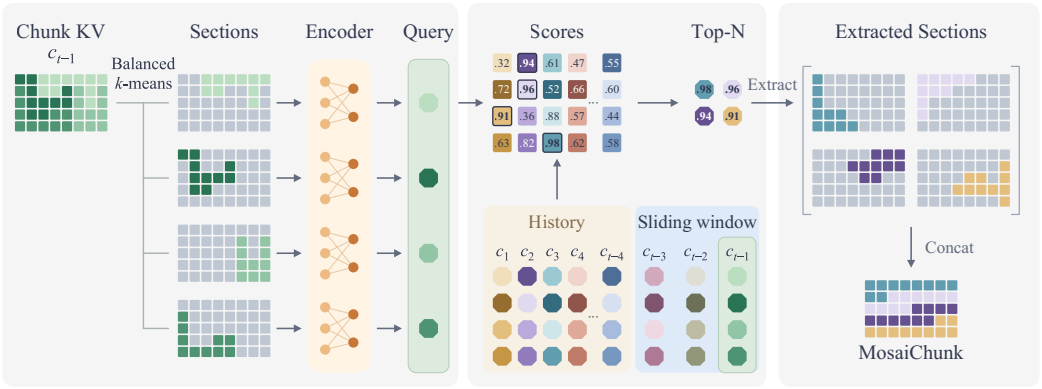

Figure 3: **Memory architecture.** (a) Each chunk is partitioned into sections and each section is encoded into a descriptor. (b) Sections in the latest chunk query history outside the sliding window. (c) The N sections with the highest scores are extracted and composed into a *MosaiChunk*.

most recent chunks, as well as a fixed attention sink used to stabilize generation; we omit this additional sink from the explanations below for simplicity.

To condition generation on more distant memory, existing retrieval methods select past chunks as additional context (Cai et al., 2025; Yi et al., 2026). But are whole chunks necessary? We test whether a video generator can be conditioned on non-contiguous chunks, providing targeted KV sections as additional context. We select only the KV entries that represent the relevant content as far memory. We ask if these entries can steer the generation and bring that content back.

Figure 2 illustrates this intervention. We retain a copy of an earlier chunk's KV before eviction and manually mark the cookie region in one frame. We map this region to latent positions and select the corresponding cached keys and values. We supply them alongside the sliding window when the tin opens again. As illustrated in the figure, the generated frame preserves the cookie's pink, star-shaped appearance, demonstrating that a frozen autoregressive video generator can directly reuse a non-contiguous subset of its own previously cached KV, even after those entries have left the active context. This observation motivates our approach, detailed below, which reuses non-contiguous KV chunks for providing video generation with rich, yet minimal, historical context.

3.2 Composing Historical KV with *MosaiChunk*

Section 3.1 shows that selecting non-contiguous historical KV entries for the relevant content can produce a consistent rollout. Our goal is to automatically identify and compose the right historical KV entries under a fixed active-cache budget. In what follows, we first describe the architecture of our memory router. We then introduce our self-distillation scheme for learning a section descriptor space in which past sections are matched and retrieved.

Memory architecture. Our *memory router* follows a three-stage design: it encodes and stores generated chunks as fine-grained sections, scores historical sections against the latest query, and composes a *MosaiChunk* from the selected sections; an overview is provided in Figure 3.

Stage 1: Encode and store. Since we want to select groups of historical KV entries with some specific semantic meaning, we partition each chunk's unrotated KV into equal-size groups using balanced k-means and call these groups *sections*. We observe that the KV entries for each section exhibit strong temporal redundancy: sections from neighboring chunks are always similar. Therefore, we train a lightweight *descriptor encoder* that maps the pooled keys of each section s to a descriptor $d(s)$. The descriptor space learns an anti-recency bias so that similarity over descriptors reflects semantic relatedness rather than temporal proximity. The *section bank* stores these sections of verbatim KV entries and their descriptors in CPU memory.

Stage 2: Score historical sections. For retrieval when generating chunk c_t, the sections of the latest generated chunk c_{t-1} form the query set $\mathcal{Q}$, and the historical sections outside the sliding window

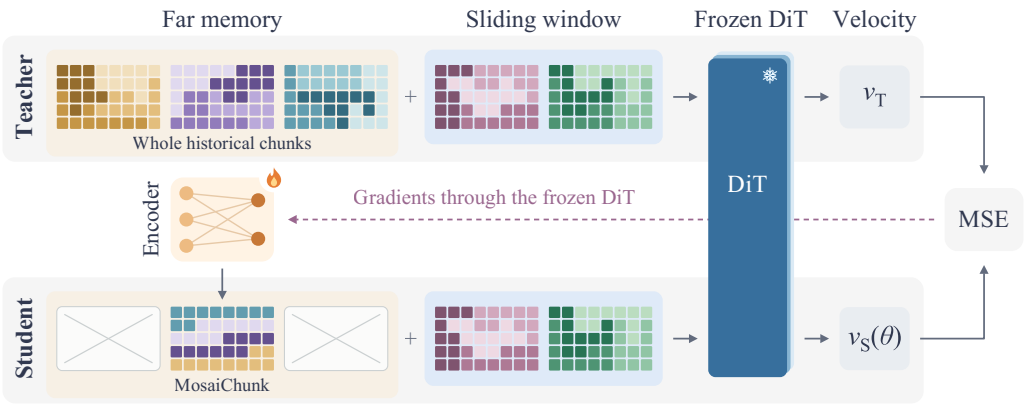

Figure 4: **Training pipeline.** The far memory consists of whole historical chunks for the teacher and a smaller *MosaiChunk* composed by our memory router for the student. The far memory is joined with the same sliding window and sent into the frozen DiT. Matching their velocity predictions updates the descriptor space of the encoder.

form the candidate set $\mathcal{H}$. The goal is to retrieve sections from $\mathcal{H}$ given all sections in $\mathcal{Q}$. We compute similarity between the descriptors of $\mathcal{Q}$ and $\mathcal{H}$. The router scores each candidate $h \in \mathcal{H}$ as:

$$\text{score}(h) = \max_{q \in \mathcal{Q}} \cos\big(d(q),\, d(h)\big), \qquad h \in \mathcal{H}. \tag{1}$$

The score measures a candidate's highest descriptor similarity to any query section. High-scoring sections therefore retrieve historical content relevant to the latest chunk.

Stage 3: Extract and compose. The router ranks all candidate sections together and selects the global top-N. Since sections have equal size, the far-memory budget determines N. A softmax over all candidate scores gives a weight for each section. We normalize the selected weights to unit mean and use them to scale the selected sections' values. We then concatenate their KV into a *MosaiChunk*, which serves as *far memory* for chunk t. The frozen DiT reads the *MosaiChunk* alongside the sliding window. We sweep multiple values of N in our evaluations, where a larger N corresponds to a larger *MosaiChunk* budget and broader coverage of the history.

Training the descriptor encoder. We train the descriptor encoder through self-distillation, teaching a student conditioned on a composed *MosaiChunk* to match the denoising velocity of the same frozen generator acting as a teacher with richer far memory that entails the historical content to be revisited.

Both the teacher and the student run one pass through the same frozen DiT with the same noisy latent and sliding window. Only the far memory differs, as shown in Figure 4. The *teacher* receives whole historical chunks that fully cover the historical content to be redrawn. We identify these chunks from the input prompt schedule for T2V models or matching input camera poses for I2V models. The *student* receives the router's *MosaiChunk* under a smaller memory budget. It must therefore select useful sections rather than copy all of the teacher's context. Let θ denote the parameters of the descriptor encoder, v_T the teacher's denoising velocity prediction, and $v_S(\theta)$ the student's prediction. We minimize their mean squared error over training chunks and noise levels:

$$\mathcal{L}_{\text{distill}}(\theta) = \mathbb{E}\left[\|v_S(\theta) - v_T\|_2^2\right]. \tag{2}$$

The teacher prediction is a fixed target. Gradients pass through the student's value weights to the descriptor encoder, not through the discrete top-N indices. The stored KV and backbone parameters remain unchanged. Since the backbone itself supplies the target, training requires no rendered ground-truth target videos.

Since teacher and student differ only in far memory, the self-distillation signal directly supervises memory selection and weighting. The teacher's whole chunks come from a much earlier visit to the same content, so matching its predictions encourages the student to recover useful details from older history. This encourages an anti-recency bias to emerge in the learned descriptor space: sections are favored for their relevant content rather than their recency alone.

Table 1: **T2V results on *RememBench*.** Median CLIP and LPIPS on 100 marked departure–revisit pairs, with rollout-quality metrics alongside. Budgets are in chunk equivalents; Base uses the same total active KV cache budget. Best CLIP and LPIPS at each budget are bold.

	$\|\mathcal{F}_c\| = 1$				$\|\mathcal{F}_c\| = 2$			
Method	CLIP ↑	LPIPS ↓	TempSSIM ↑	Drift ↓	CLIP ↑	LPIPS ↓	TempSSIM ↑	Drift ↓
Base	0.755	0.659	0.923	0.051	0.751	0.652	0.926	0.050
MoC	0.832	0.594	0.928	0.052	0.841	0.569	0.930	0.053
Ours	**0.899**	**0.567**	0.932	0.052	**0.936**	**0.500**	0.932	0.055

4 BENCHMARKS

Existing benchmarks such as WBench (Ying et al., 2026), WorldMark (Xu et al., 2026), and PersistBench (He et al., 2026) evaluate video quality and consistency, but do not systematically test *long-horizon revisits* in which frames establishing the target content's appearance are evicted from the model's sliding window. We therefore introduce *RememBench*, a benchmark with two splits for evaluating consistency across such revisits in autoregressive video models. Revisits are driven by prompts in the T2V split and by camera motion in the I2V split. Additional benchmark construction details and first-frame visualizations are provided in Supplementary Sections B.1 and B.2, respectively.

The T2V split. The split contains 100 samples with scenarios disjoint from router training. Each model input extends Ring Forcing's three-stage appear–disappear–reappear design (Xue et al., 2026) to four prompt segments, with a separate segment keeping the object out of sight.

The I2V split. The split contains 150 scenes: 50 indoor and 100 outdoor. Each model input includes an initial frame from DL3DV (Ling et al., 2023), a prompt describing the scene, and a camera trajectory. We sample more diverse camera trajectories not seen during training: all 150 scenes have in-place rotation trajectories with $90°$, $180°$, and $360°$ settings. The 100 outdoor scenes additionally have trajectories combining these rotations with translation.

5 RESULTS

We evaluate whether *MosaiChunk* provides useful fine-grained historical information beyond a sliding window and compare it with existing retrieval mechanisms that condition on whole chunks, testing whether section-level memory better preserves visual details. We conduct both comparisons on the T2V and I2V splits of *RememBench*, using the frozen H3-AR backbone for T2V and LingBot-World-Infinity for I2V. Video comparisons are available in the accompanying `video.html`.

5.1 EVALUATION PROTOCOL

Baseline. We compare *MosaiChunk* with the sliding-window baseline (Base) and Mixture of Contexts (MoC) (Cai et al., 2025), a whole-chunk retrieval mechanism. We adapt MoC on each frozen backbone. Across baselines, T2V runs share prompts and noise seeds; I2V runs share conditioning frames and camera trajectories.

Notation. Let $\mathcal{F}_c$ denote the far memory used to generate chunk c, and $\|\mathcal{F}_c\|$ its budget in video-chunk equivalents. For example, $\|\mathcal{F}_c\| = 1$ means that the size of the far memory is equivalent to one video chunk. We evaluate budgets of one and two chunks.

Metric. CLIP similarity and LPIPS assess consistency between departure and revisit frames. T2V pairs are manually marked; I2V pairs use the conditioning frame and a revisit selected from Pi3X-reconstructed camera poses (Wang et al., 2026). TempSSIM (consecutive-frame SSIM (Wang et al., 2004)) and Local Scene Drift (Drift) (Wu et al., 2026) measure rollout quality. We report medians over scenes. More details are provided in Supplementary Material Section B.3.

Table 2: **I2V results on *RememBench*.** Median revisit CLIP and LPIPS for 180° turns: 150 scenes with rotation and 100 with translation. Budget and bolding conventions follow Table 1.

	$\|\mathcal{F}_c\| = 1$				$\|\mathcal{F}_c\| = 2$			
Method	CLIP ↑	LPIPS ↓	TempSSIM ↑	Drift ↓	CLIP ↑	LPIPS ↓	TempSSIM ↑	Drift ↓
Rotation								
Base	0.768	0.655	0.457	0.051	0.793	0.650	0.454	0.052
MoC	0.807	0.645	0.457	0.050	0.807	0.642	0.460	0.049
Ours	**0.840**	**0.625**	0.456	0.051	**0.863**	**0.609**	0.458	0.051
Rotation + translation								
Base	0.788	0.625	0.527	0.043	0.813	0.611	0.521	0.044
MoC	0.812	0.617	0.521	0.045	0.830	0.616	0.521	0.046
Ours	**0.848**	**0.606**	0.519	0.044	**0.862**	**0.584**	0.513	0.046

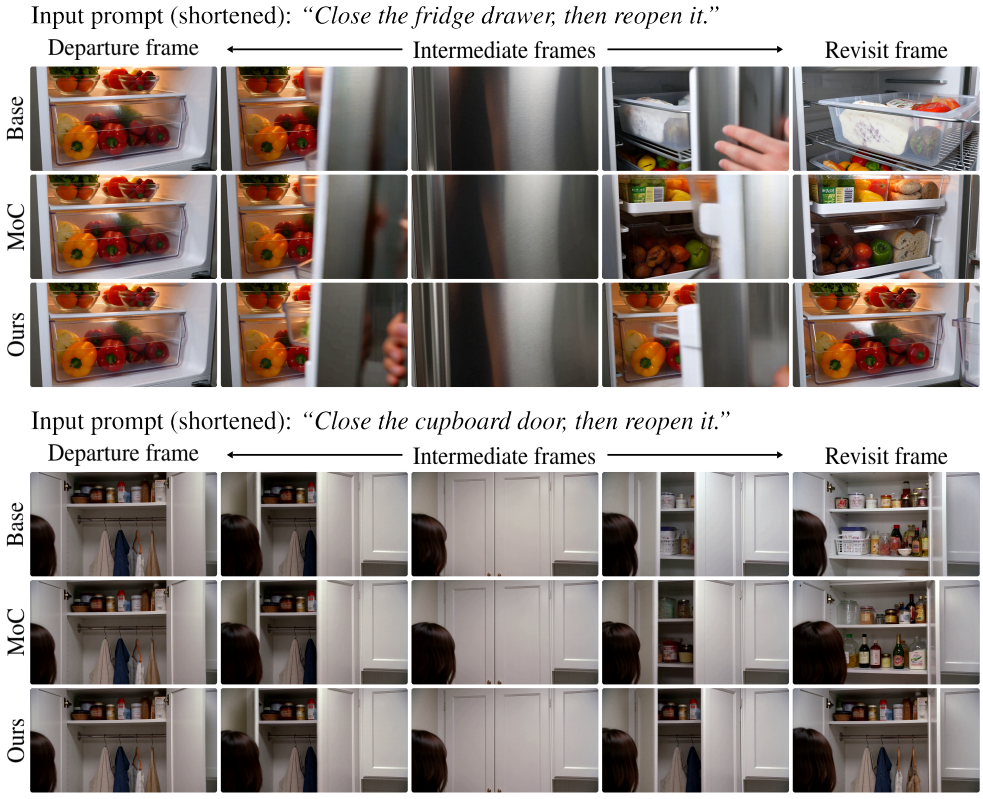

Figure 5: **T2V revisits at a two-chunk far-memory budget.** Rows (Base, MoC, and Ours) share the prompt and noise within each scene. The first and last columns show the departure and revisit frames selected for evaluation. Only *MosaiChunk* preserves the vegetables and their arrangement (top) and the cupboard's contents (bottom).

5.2 Text-to-Video Results

Table 1 shows that *MosaiChunk* improves median CLIP over Base by 0.144 and 0.185 at budgets of one and two chunks, respectively, and over MoC by 0.067 and 0.095. LPIPS shows the same ordering. These results support allocating the retrieval budget to content-relevant sections rather than whole chunks. At each budget, TempSSIM and Drift differ by less than 0.01 across methods, indicating similar rollout quality under these metrics. Figure 5 illustrates these differences at the two-chunk budget. Only *MosaiChunk* preserves the vegetables and their arrangement after the refrigerator reopens, and retains the cupboard's contents after its doors reopen.

Table 3: **I2V results on public benchmarks.** WBench uses the navigation split; WorldMark uses the first-person splits. Budgets specify *MosaiChunk*'s far memory in chunk equivalents; Base matches the total active KV cache budget. Higher is better for all metrics; bold marks the better consistency score at each budget.

		WBench						WorldMark		
		Consistency				Video quality		Consistency	Video quality	
$\|\mathcal{F}_c\|$	Method	Spatial	Gated Spatial	Geom.	Subject	Aesth.	Imaging	Revisit Memory	Aesth.	Percept.
1	Base	78.28	75.47	85.98	88.13	61.75	67.31	80.14	61.43	85.49
	Ours	**79.63**	**76.69**	**86.74**	**88.36**	61.86	67.42	**84.89**	62.48	86.45
2	Base	79.43	76.78	86.14	88.10	61.76	67.40	84.63	61.05	84.97
	Ours	**80.73**	**78.18**	**87.08**	**88.38**	61.92	67.45	**84.87**	62.89	86.64

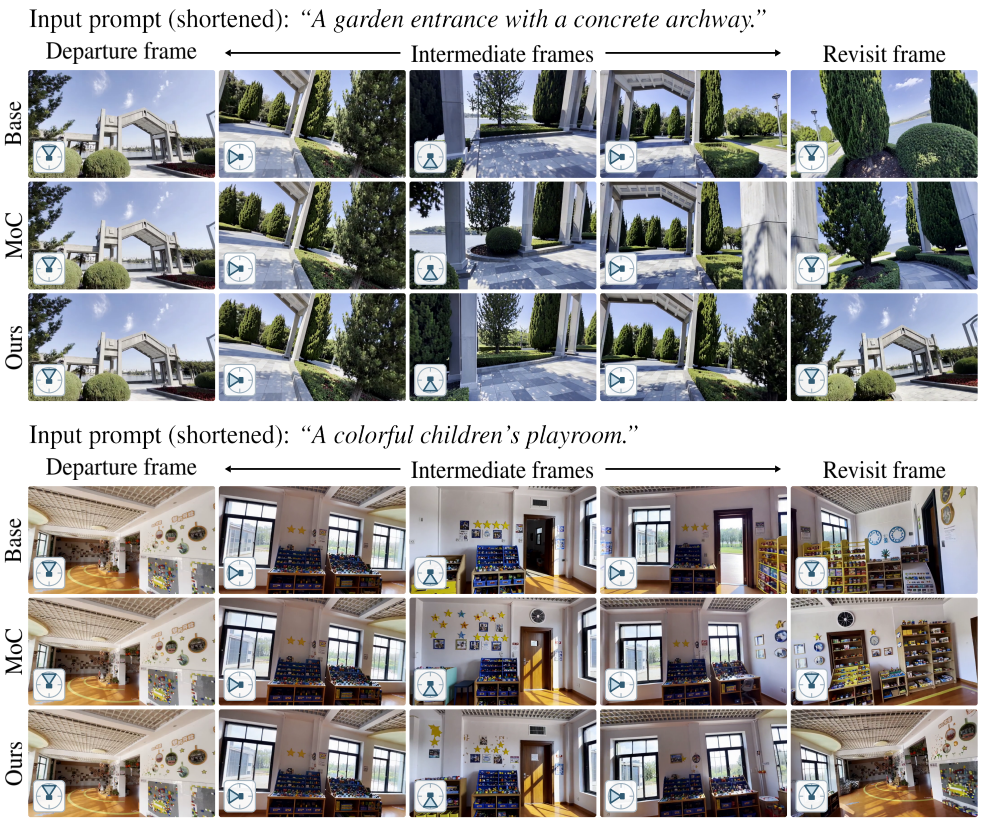

Figure 6: **I2V revisits at two-chunk (top) and one-chunk (bottom) far-memory budgets.** Rows (Base, MoC, and Ours) share the conditioning frame and 180° camera trajectory within each scene. The first column shows the departure frame; the last shows each method's revisit frame, selected from its Pi3X reconstruction. Intermediate columns sample the camera turn; camera insets schematically show the commanded rotation. *MosaiChunk* better recovers the garden entrance's archway (top) and the playroom's layout and wall decorations (bottom).

5.3 Image-to-Video Results

Table 2 reports 180° turns with and without translation. At a one-chunk budget, *MosaiChunk* improves median CLIP over Base by 0.072 under rotation and 0.060 with translation, and over MoC by 0.033 and 0.036. At two chunks, the corresponding gains are 0.070 and 0.049 over Base, and 0.056 and 0.032 over MoC. LPIPS again improves in all four settings. At each budget and trajec-

tory, TempSSIM and Drift differ by less than 0.01 across methods. Results for $90°$ and $360°$ turns are provided in Supplementary Material Section C.2. Figure 6 shows two $180°$ examples in which *MosaiChunk* more faithfully recovers the garden entrance's archway and the playroom's layout and wall decorations than either baseline.

In addition, we evaluate I2V generation on two public benchmarks, WBench (Ying et al., 2026) and WorldMark (Xu et al., 2026) (Table 3). At both budgets, *MosaiChunk* improves every consistency score in Table 3 over the sliding-window baseline while maintaining video quality. Full results and metric descriptions are provided in Supplementary Material Section C.2.

5.4 Ablation Studies

We ablate two key design choices in our memory architecture: (1) learning a descriptor space for retrieval and (2) selecting sections globally under a shared memory budget. All variants use the same section partition and frozen backbone, evaluated on 150 I2V rotation scenes at one- and two-chunk far-memory budgets (Table 4).

Table 4: **Memory architecture ablation.** Median revisit CLIP and LPIPS on 150 I2V rotation scenes. Disabling descriptor learning uses mean-pooled keys; disabling global selection uses per-query top-k. The last row is the full *MosaiChunk* router, with results from Table 2.

Router design		$\|\mathcal{F}_c\| = 1$		$\|\mathcal{F}_c\| = 2$	
Learned descriptors	Global top-N	CLIP ↑	LPIPS ↓	CLIP ↑	LPIPS ↓
✘	✔	0.815	0.637	0.841	0.640
✔	✘	0.792	0.656	0.792	0.655
✔	✔	**0.840**	**0.625**	**0.863**	**0.609**

Learning the descriptor space. As discussed in Section 3.2, temporally neighboring sections have similar keys. Directly matching pooled keys can therefore favor recent sections over older content needed at a revisit. We train the descriptor encoder to counter this recency bias. We replace learned descriptors with mean-pooled keys and score their cosine similarity, keeping global top-N selection and using uniform value weights. The learned router achieves higher CLIP and lower LPIPS at both budgets.

Selecting sections under a shared budget. We retrain the router with MoC-style per-query top-k selection (Cai et al., 2025) under the same memory budget. Intuitively, content that never leaves view may remain in the sliding window, so retrieving k sections for every query can spend budget on information the backbone already has; global selection instead devotes the budget to relevant content missing from local context. Global top-N again improves both metrics at both budgets, making the full design the best-performing variant in Table 4.

Since camera poses are available in I2V, we also test a pose-guided variant of our router, following pose-based retrieval methods (Yi et al., 2026). We further evaluate fractional far-memory budgets of 0.5 and 1.5 chunks and report the budget curve. Both studies are provided in Supplementary Material Section C.3.

6 Conclusion

We presented *MosaiChunk*, a spatio-temporal memory mechanism for preserving visual content across long-horizon revisits, and *RememBench* for evaluating this capability. Our self-distillation scheme trains the router using the same frozen backbone with richer historical context as the teacher. While the router improves visual memory, generation remains constrained by the teacher's capabilities. Because the backbone's weights remain fixed, our method inherits its limitations in instruction following and its learned generative distribution. Future video models could instead build memory directly into the generator, learning during pre-training or post-training to natively reuse visual content from earlier in the generated history, even after it has left the active context.

AI USE STATEMENT

We used generative AI tools to assist with data construction, research-code implementation and debugging, interpreting experimental results, and manuscript preparation. Specifically for data generation, a Claude agent wrote T2V prompt scenarios, while Qwen2.5-VL-7B assisted with I2V scene filtering and captioning, as described in Supplementary Section B.1. An author manually reviewed all AI-assisted text, code, data, and output visualizations. The authors remain responsible for the accuracy and integrity of the methods, results, and conclusions reported in this work.

ETHICS STATEMENT

MosaiChunk trains only a lightweight memory router while keeping the pretrained video backbone frozen. It selects and composes historical KV entries without updating the generator's parameters. The method therefore inherits limitations of its pretrained backbone and source data, including potential biases and harmful behaviors. We do not identify additional ethical concerns specific to the proposed memory router.

REPRODUCIBILITY STATEMENT

Section 3 describes the memory architecture and the self-distillation objective. Supplementary Sections A.1 and A.2 provide implementation and training details, including hyperparameters and hardware. Supplementary Section B.1 documents the data sources, filtering procedures, prompts, and camera trajectories used to construct *RememBench*, while Section B.3 specifies generation settings, baseline adaptations, matched memory budgets, revisit-pair selection, and metrics. We will publicly release our code, data, and trained router checkpoints to facilitate reproduction and further research.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Oren Boiman and Michal Irani. Similarity by Composition. In *Neural Information Processing Systems*, pp. 177–184, 2006. URL https://mlanthology.org/neurips/2006/boiman2006neurips-similarity/.

Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyan Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, Maneesh Agrawala, Lu Jiang, and Gordon Wetzstein. Mixture of contexts for long video generation, 2025. URL https://arxiv.org/abs/2508.21058.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion, 2024. URL https://arxiv.org/abs/2407.01392.

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, 2026. URL https://arxiv.org/abs/2601.21896.

Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of SIGGRAPH*, 2001. URL https://people.eecs.berkeley.edu/~efros/research/quilting.html.

Gal Fiebelman, Hadar Averbuch-Elor, and Sagie Benaim. Mv-forcing: Long multi-view video generation via 4d-grounded spatio-temporal self-forcing. *arXiv preprint arXiv:2607.05376*, 2026.

Yizhao Gao, Zhichen Zeng, DaYou Du, Shijie Cao, Peiyuan Zhou, Jiaxing Qi, Junjie Lai, Hayden So, Ting Cao, Fan Yang, and Mao Yang. SeerAttention: Self-distilled attention

gating for efficient long-context prefilling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 55846–55869. Curran Associates, Inc., 2025. doi: 10.52202/085713-1869. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/50e9dbc4ab68d94f15261ddc26c8ca2b-Paper-Conference.pdf.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, Ka Leong Cheng, Haojie Zhang, Jian Gao, Tianrui Feng, Yuzheng Liu, Yao Yao, Yinghao Xu, Xing Zhu, Yujun Shen, and Hao Ouyang. Infinite worlds with versatile interactions, 2026. URL https://arxiv.org/abs/2607.07534.

James Hays and Alexei A. Efros. Scene completion using millions of photographs. *ACM Trans. Graph.*, 26(3):4–es, July 2007. ISSN 0730-0301. doi: 10.1145/1276377.1276382. URL https://doi.org/10.1145/1276377.1276382.

Guangzhao He, Hadar Averbuch-Elor, and Wei-Chiu Ma. Can 4d foundation models remember?, 2026. URL https://arxiv.org/abs/2609.20819.

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, Kalyan Sunkavalli, Feng Liu, Zhengqi Li, and Hao Tan. Relic: Interactive video world model with long-horizon memory, 2025. URL https://arxiv.org/abs/2512.04040.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion, 2025. URL https://arxiv.org/abs/2506.08009.

Phillip Isola and Ce Liu. Scene collaging: Analysis and synthesis of natural images with semantic layers. In *ICCV*, 2013.

Yicheng Ji, Zhizhou Zhong, Jun Zhang, Qin Yang, XiTai Jin, Ying Qin, Wenhan Luo, Shuiyang Mao, Wei Liu, and Huan Li. Forcing-kv: Hybrid kv cache compression for efficient autoregressive video diffusion models, 2026. URL https://arxiv.org/abs/2605.09681.

Nebojsa Jojic, Brendan Frey, and Anitha Kannan. Epitomic analysis of appearance and shape. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 34–41 vol.1, 2003. doi: 10.1109/ICCV.2003.1238311.

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

Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, Zizhen Li, Fanrui Zhang, Jiaxin Ai, Zhixiang Wang, Yuwei Wu, Tong He, Jiangmiao Pang, Yu Qiao, Yunde Jia, and Kaipeng Zhang. Sekai: A video dataset towards world exploration, 2025. URL https://arxiv.org/abs/2506.15675.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views, 2025. URL https://arxiv.org/abs/2511.10647.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision, 2023. URL https://arxiv.org/abs/2312.16256.

Yanzuo Lu, Ronglai Zuo, and Jiankang Deng. Raven: Real-time autoregressive video extrapolation with consistency-model grpo, 2026. URL https://arxiv.org/abs/2605.15190.

Jiayi Luo, Qiyan Liu, Tengyang Wang, JunHao Liu, Jiayu Chen, Cong Wang, Hanxin Zhu, Chen Gao, Xiaobin Hu, Qingyun Sun, and Zhibo Chen. Future forcing: Future-aware training-free kv cache policy for autoregressive video generation, 2026. URL https://arxiv.org/abs/2605.30083.

MiniMax. MiniMax H3: An open model breaking the boundaries between tasks and modalities. https://www.minimax.io/blog/minimax-h3, 2026. Model weights: https://huggingface.co/MiniMaxAI/MiniMax-H3.

Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video textures. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pp. 489–498, USA, 2000. ACM Press/Addison-Wesley Publishing Co. ISBN 1581132085. doi: 10.1145/344779.345012. URL https://doi.org/10.1145/344779.345012.

J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman. Discovering objects and their location in images. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pp. 370–377 Vol. 1, 2005. doi: 10.1109/ICCV.2005.77.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning, 2026. URL https://arxiv.org/abs/2507.13347.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Ydo Wexler, Eli Shechtman, and Michal Irani. Space-Time Video Completion. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pp. 120–127, Los Alamitos, CA, USA, 2004. IEEE Computer Society. doi: 10.1109/CVPR.2004.1315022. URL https://www.microsoft.com/en-us/research/publication/space-time-video-completion/.

Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory, 2025a. URL https://arxiv.org/abs/2506.05284.

Xiaofei Wu, Guozhen Zhang, Zhiyong Xu, Yuan Zhou, Qinglin Lu, and Xuming He. Pack and force your memory: Long-form and consistent video generation, 2025b. URL https://arxiv.org/abs/2510.01784.

Xindi Wu, Sven Elflein, James Lucas, Olga Russakovsky, Laura Leal-Taixé, Despoina Paschalidou, Jonathan Lorraine, and Aljoša Ošep. Addressable memory for video world models, 2026. URL https://arxiv.org/abs/2608.07408.

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, 2026. URL https://arxiv.org/abs/2602.02958.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory, 2025. URL https://arxiv.org/abs/2504.12369.

Xiaojie Xu, Zhengyuan Lin, Kang He, Yukang Feng, Xiaofeng Mao, Yuanyang Yin, Yongtao Ge, and Kaipeng Zhang. Worldmark: A unified benchmark suite for interactive video world models. *arXiv preprint arXiv:2604.21686*, 2026.

Bowen Xue, Brandon Y. Feng, Chenguo Lin, Yuchen Lin, Yujia Zeng, Lvmin Zhang, Maneesh Agrawala, Honglei Yan, and Panwang Pan. Ring forcing: Towards precise long-term memory for autoregressive video diffusion, 2026. URL https://arxiv.org/abs/2608.26794.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. Longlive: Real-time interactive long video generation, 2025. URL https://arxiv.org/abs/2509.22622.

Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression, 2025. URL https://arxiv.org/abs/2512.05081.

Jung Yi, Minjae Kim, Paul Hyunbin Cho, Wooseok Jang, Sangdoo Yun, and Seungryong Kim. Worldkv: Efficient world memory with world retrieval and compression, 2026. URL https://arxiv.org/abs/2605.22718.

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, 2025. URL https://arxiv.org/abs/2412.07772.

Kaining Ying, Hengrui Hu, Siyu Ren, Jiamu Li, Fengjiao Chen, Ziwen Wang, Xuezhi Cao, Xunliang Cai, and Henghui Ding. Wbench: A comprehensive multi-turn benchmark for interactive video world model evaluation. *arXiv preprint arXiv:2605.25874*, 2026.

Jiwen Yu, Jianhong Bai, Yiran Qin, Quande Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. Context as memory: Scene-consistent interactive long video generation with memory retrieval, 2025. URL https://arxiv.org/abs/2506.03141.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models, 2025. URL https://arxiv.org/abs/2504.12626.

Lin Zhao, Yushu Wu, Yifan Gong, Yanzhi Wang, and Pu Zhao. Omnimem: Scalable and adaptive memory retrieval for long video generation, 2026. URL https://arxiv.org/abs/2605.30519.

Supplementary Material

We refer readers to the interactive visualizations in the accompanying `video.html`, which show text-to-video (T2V) and image-to-video (I2V) results. In this document, we provide implementation details (Section A), describe benchmark construction, scene visualizations, and evaluation protocols (Section B), and present additional results (Section C).

Contents

A Implementation Details **14**
A.1 Memory Router 14
A.2 Self-Distillation 15
B Benchmark Details **17**
B.1 Dataset Construction 17
B.2 Visualization of First Frames 18
B.3 Detailed Evaluation Protocol 18
C Additional Results **21**
C.1 Extra Qualitative Results 21
C.2 Extra Quantitative Results 23
C.3 Extra Ablations 25
C.4 KV Transplantation 27

A Implementation Details

MosaiChunk adds a trainable memory router to a frozen autoregressive video generator. The router composes historical KV sections into a *MosaiChunk* under a fixed far-memory budget. The frozen generator reads it alongside a sliding window of two recent chunks and a fixed *attention sink*. We use RAVEN-adapted MiniMax-H3 (H3-AR) for text-to-video (T2V) and LingBot-World-Infinity for image-to-video (I2V) (MiniMax, 2026; Lu et al., 2026; Gao et al., 2026). We observed content leakage through a visual sink in H3-AR, but not in LingBot-World-Infinity. We therefore retain text tokens as the attention sink for H3-AR and the first video chunk's tokens for LingBot-World-Infinity.

A.1 Memory Router

Stage 1: Encode and store. We partition each chunk's unrotated KV into equal-size groups, called *sections*, using balanced k-means. Equal section sizes let the number of selected sections determine the KV budget. Clustering uses keys before rotary positional encoding (RoPE), at layer 25 of H3-AR's 50 layers and layer 20 of LingBot-World-Infinity's 40 layers. The same section partition applies to all layers; sections may span non-contiguous positions and multiple frames. Table 5 lists their sizes and the layers used for descriptors.

Table 5: **Memory router settings for Stage 1.** Token counts refer to visual tokens only; layer indices are zero-based. Keys from the clustering layer are used to partition each chunk into equal-size sections. Pooled keys from the descriptor layers are used to compute each section's descriptor.

	T2V (H3-AR)	I2V (LingBot-World-Infinity)
Transformer layers	50	40
Attention heads $\times$ head width	56×128	40×128
Key width across heads	7168	5120
Latent frames per chunk	5	4
Spatial patch grid	24×43	30×52
Visual tokens per chunk	5160	6240
Sections per chunk, S	40	48
Tokens per section	129	130
Clustering layer	25	20
Descriptor layers	$12, 25, 38$	$10, 20, 30$

To compute each section's descriptor, we pool its unrotated keys at layers $12, 25, 38$ for H3-AR and $10, 20, 30$ for LingBot-World-Infinity. Each token's keys across attention heads form one vector.

Elementwise mean, maximum, and minimum over the section's tokens capture average responses and extremes that averaging would lose (Gao et al., 2025). These three statistics at three layers yield nine pooled key vectors. The *descriptor encoder* normalizes their scales, projects them to width 1024, adds embeddings identifying the source layer and statistic, and refines them with a shared residual MLP (hidden width 2048, GELU). Separate output projections give query and key descriptor sets $d_Q(s)$ and $d_K(s)$, each with nine vectors, so the same section can serve either role in comparison.

To keep full historical KV off the GPU, the *section bank* stores verbatim KV entries from all backbone layers in bfloat16 on the CPU, together with each section's token indices. Stored history grows with the rollout; the DiT's active KV budget remains fixed.

Stage 2: Score historical sections. When generating chunk t, sections of the latest generated chunk $t-1$ form the query set $\mathcal{Q}$. Historical sections outside the sliding window and attention sink form the candidate set $\mathcal{H}$.

Before comparison, we center key descriptor vectors within $\mathcal{H}$ for each source layer and pooling statistic, to remove components shared across sections. We normalize query and centered key vectors to unit length for cosine similarity. For descriptor sets A and B, $\mathrm{sim}(A, B)$ averages the best cosine match in B for each of A's nine vectors. Each candidate receives the score

$$s(h) = \max_{q \in \mathcal{Q}} \mathrm{sim}\big(d_Q(q), d_K(h)\big), \quad h \in \mathcal{H}. \tag{3}$$

This favors sections that match a current query while complementing the content that is missing from the sliding window.

Stage 3: Extract and compose. We rank all candidate sections by their scores and select as many as the far-memory budget allows. One- and two-chunk budgets allow 40 and 80 sections for T2V, and 48 and 96 for I2V, respectively. Their stored token indices retrieve the verbatim KV entries at every backbone layer.

Softmax converts all candidate scores into weights, giving higher-scoring sections larger weights. We normalize the selected weights to average 1 and use them to scale each section's values. This differentiable weighting lets the denoising loss train the descriptor encoder through the frozen DiT.

We preserve each selected token's spatial coordinates and adjust its temporal RoPE coordinates to place it before the sliding window, fitting the backbone's context layout. For N selected sections, let K_i, V_i denote section i's KV after this positional adjustment, and w_i its value weight. At each backbone layer, the ***MosaiChunk*** keys and values are

$$\begin{aligned} K_{\text{far}} &= \mathrm{Concat}(K_1, \ldots, K_N), \\ V_{\text{far}} &= \mathrm{Concat}(w_1 V_1, \ldots, w_N V_N). \end{aligned} \tag{4}$$

Here Concat joins token rows. The frozen DiT reads the resulting ***MosaiChunk*** as far memory alongside the attention sink and sliding window.

A.2 Self-Distillation

We train the descriptor encoder through self-distillation to match the same frozen generator's predictions under richer far memory. Teacher and student use the same frozen DiT, noisy latent, denoising step, and sliding window; only far memory differs.

Teacher and student. The KV context consists of an attention sink, a sliding window, and far memory. We specify these components for each task below.

T2V. Both teacher and student use text tokens as the attention sink and the two most recent video chunks as the sliding window. The teacher keeps the first five video chunks, which cover the prompt's initial reveal of the target content. Their KV outside the sliding window forms its far memory. The student's far memory is a ***MosaiChunk*** composed from historical sections outside the sliding window, under a two-chunk far-memory budget.

I2V. Both teacher and student use the first video chunk as the attention sink and the two most recent chunks as the sliding window. The teacher's far memory contains up to three historical chunks

nearest to the current camera pose, selected from outside the attention sink and sliding window. For the in-place training trajectory, nearness is measured by the angle between chunks' mean viewing directions. The student's far memory is a *MosaiChunk* composed from historical sections outside the attention sink and sliding window, under a two-chunk far-memory budget.

We minimize the mean squared error (MSE) between teacher and student predictions, comparing noise-free latents for H3-AR and denoising velocities for LingBot-World-Infinity. We scale both losses by 100. For T2V, we first generate a 16-chunk rollout, then uniformly sample one chunk from indices 5–15 and one of its four denoising steps for supervision. For I2V, we generate 20 chunks sequentially. For each chunk at indices 4–19, we uniformly sample one of four denoising steps and update the descriptor encoder at that step before continuing generation. Each comparison uses the same noisy latent for teacher and student. Chunk indices are zero-based.

Teacher predictions are fixed targets. Gradients pass through the student's frozen DiT and the section value weights to the descriptor encoder; they do not pass through the discrete section selection. The stored KV and backbone parameters remain unchanged. The mean used to normalize value weights is held constant during backpropagation, preserving gradients through the full-candidate softmax to unselected scores as well. To explore alternatives during training, random candidates replace 15% of selected sections for T2V and a fraction drawn uniformly from $[0, 0.15]$ for I2V. Evaluation disables this exploration.

Training data. T2V uses 2000 four-segment prompt sequences based on Ring Forcing's three-stage appear–disappear–reappear design (Xue et al., 2026), with 16 chunks per rollout. I2V starts from 2664 Sekai clips (Li et al., 2025), captioned with Qwen2.5-VL-7B (Bai et al., 2025). Scene-disjoint splits yield 2388 training clips and $44/46$ validation/test clips after capping each held-out scene at three clips. Sekai supplies conditioning frames; the frozen backbone generates continuations, so no ground-truth target videos are required. Training uses a 20-chunk in-place turn of roughly $120°$ followed by its reversal. The DL3DV evaluation scenes and $90°$, $180°$, $360°$, and translation trajectories are separate from this training setup (Section B.1).

Optimization. Table 6 reports architecture, optimization, and checkpoint settings. Each backbone has its own trained router, reused across memory budgets. Scheduled steps specify the training horizon; evaluation uses the earlier checkpoint listed in the table.

Table 6: **Self-distillation configuration.** Only the router is optimized. Scheduled steps describe the configured training schedule; reported checkpoints identify the weights used for evaluation.

	T2V (H3-AR)	I2V (LingBot-World-Infinity)
Hardware	32 NVIDIA H200	16 NVIDIA H200
Parallelism	FSDP, sequence parallel 4	DDP
Sequences per step	8	16
Trainable parameters	13.65 M	11.55 M
Optimizer	AdamW	AdamW
Learning rate	10^{-4}	2×10^{-4}
Adam betas	$(0.9, 0.95)$	$(0.9, 0.999)$
Weight decay	0.01	0
Learning-rate schedule	Constant	200-step warmup, then cosine
Gradient clipping	1.0	1.0
Scheduled steps	4000	4768 (two epochs)
Reported checkpoint	Step 416	Step 500
Training data	2000 prompt sequences	2388 Sekai clips
Training rollout	16 chunks	20 chunks
Student far memory	2 chunks	2 chunks
Sliding window	2 video chunks	2 video chunks
Attention sink	Text tokens	First video chunk
Teacher historical chunks	First 5 video chunks	3 chunks with closest poses
Descriptor width	1024	1024
Initial score multiplier, β	8 (fixed)	8 (learned)



Figure 7: **T2V prompt-compliance failures.** Each row shows a discarded scene under sliding-window inference. The upper band marks the prompt transitions and sampled instants. Top: the container closes but never reopens. Bottom: the container never closes.

B Benchmark Details

This section describes the construction and evaluation of *RememBench* and illustrates its visual diversity. T2V test scenarios are disjoint from router training. I2V uses initial frames from DL3DV rather than the Sekai frames used for training, together with camera trajectories not seen during training.

B.1 Dataset Construction

T2V prompts and screening. We adapt Ring Forcing's three-stage appear–disappear–reappear design (Xue et al., 2026) into four prompt segments with a Claude agent: the object appears, disappears, remains out of sight, and reappears. The candidate pool contains 466 scenes covering cases, drawers, doors, lids, covers, tins, hinges, latches, screw tops, and sliding covers. The nominal prompt transitions occur at 3.6, 6.2, and 11.2 seconds. The third segment therefore requests five seconds with the object out of sight, exceeding the largest sliding-window baseline's approximately three seconds of recent context. Each generated clip contains 379 frames at 24 fps; the prompt schedule specifies the intended action timing, not the exact frame at which a generated action occurs.

For each prompt, we generate one H3-AR rollout and screen it for compliance with the four segments. We retain 100 scenes, excluding 366. A rollout in which the object never disappears or never returns has no valid departure–revisit pair and does not test the memory behavior of interest. Figure 7 shows these two failure modes. The retained scene list and noise seeds are fixed across the methods and budgets being compared.

I2V initial frames and prompts. We take initial frames from DL3DV (Ling et al., 2023), center-crop them to the backbone's aspect ratio, and resize them to 832×480. Qwen2.5-VL-7B (Bai et al., 2025) is used for scene classification, view screening, and captioning. The classification pass separates indoor and outdoor scenes. The screening pass checks eye-level views in both classes and overhead cover in outdoor scenes only. A covered outdoor scene is eligible for rotation alone; it is not used for translation. Ambiguous binary replies, containing both or neither expected answer, are not assigned a default class. The final set contains 50 indoor and 100 outdoor scenes; the selected outdoor scenes support both trajectory types in Table 8. The captioning pass produces a scene prompt describing visible content without prescribing camera motion, so the camera trajectory is supplied through the backbone's motion conditioning rather than through the text prompt. Table 7 lists the scene-classification, eye-level screening, and captioning prompts.

Table 7: **Prompts for I2V scene preparation.** These prompts cover scene classification, eye-level screening, and captioning for both indoor and outdoor frames.

Purpose	Prompt
Scene class	Is this an indoor scene or an outdoor scene? Answer with one word: indoor or outdoor.
Eye-level view	Is this photo taken from roughly a standing person's eye level, looking horizontally – not from high above the scene, not tilted down at the ground, not tilted up at the sky? Answer with one word: yes or no.
Scene prompt	Describe this scene in two or three sentences, as a caption for the image. Name the place, the main structures and surfaces, the materials, the lighting and the weather. Do NOT describe any camera motion, and do not say 'the camera' or 'the video' – describe only what is visible.

I2V camera trajectories. All 150 scenes have in-place rotation trajectories, with a left or right direction drawn once per scene and shared across settings. For 90° and 180°, yaw increases linearly to the specified angle at the midpoint and then reverses to its initial value. The 360° trajectory instead completes one continuous full turn.

For the 100 outdoor scenes, Depth Anything 3 (Lin et al., 2025) estimates depth from the initial frame. We use conservative free-depth estimates around the forward direction to define a scene-specific straight path into the scene and back. The translation distance is bounded by the estimated free space. Rotation is superimposed on this out-and-back translation, using the same three angular settings. Each input camera trajectory ends at its initial position and orientation.

B.2 Visualization of First Frames

Figure 8 shows objects when first revealed in T2V rollouts and the conditioning frames for I2V, illustrating the visual diversity of the two splits.

B.3 Detailed Evaluation Protocol

Generation and matched budgets. To isolate memory quality, all methods use the same prompts and noise seed for each scene, with a shared prompt schedule for T2V and the same conditioning frame and input camera trajectory for I2V. T2V generates 379 frames at 24 fps (23 chunks); I2V generates 253 frames at 16 fps (16 chunks). Both use four denoising steps per chunk. Retrieval methods retain two chunks in the sliding window plus one or two chunks of far memory. Base instead retains three or four recent chunks, matching the active visual KV budget. All methods use the same backbone-specific attention sink. Scores are computed on the original rollouts, before the compression used to package the accompanying `video.html`.

MoC adaptation. We adapt MoC (Cai et al., 2025) to our frozen backbones, using mean-pooled keys to describe and retrieve historical chunks. We use the latest chunk's mean-pooled section keys as retrieval queries. Each historical chunk is described by its mean-pooled keys and scored by its maximum dot product with these queries. The highest-scoring one or two chunks form far memory, with the same selection shared across queries, attention heads, and layers. Retrieved values have unit weights, and positional handling is the same as for *MosaiChunk*.

T2V frame pairs. For each retained scene, we manually mark a departure frame in which the object is fully visible before the container closes. Its frame index is shared across methods. We then mark each rollout's revisit frame separately, when the container has reopened and its contents are visible. This accommodates differences in generated action timing. The annotation interface in Figure 9 uses linked playback to mark the common departure frame. We then select and scrub each rollout individually to mark its revisit frame; the saved frame pairs appear below.

I2V frame pairs. The departure frame is the conditioning frame. We reconstruct each generated rollout with Pi3X (Wang et al., 2026) in a single pass, using every decoded frame. Reconstruction inputs are resized to approximately 255,000 pixels, with dimensions rounded to multiples of 14. The reconstructed poses determine the revisit frame independently for each method.

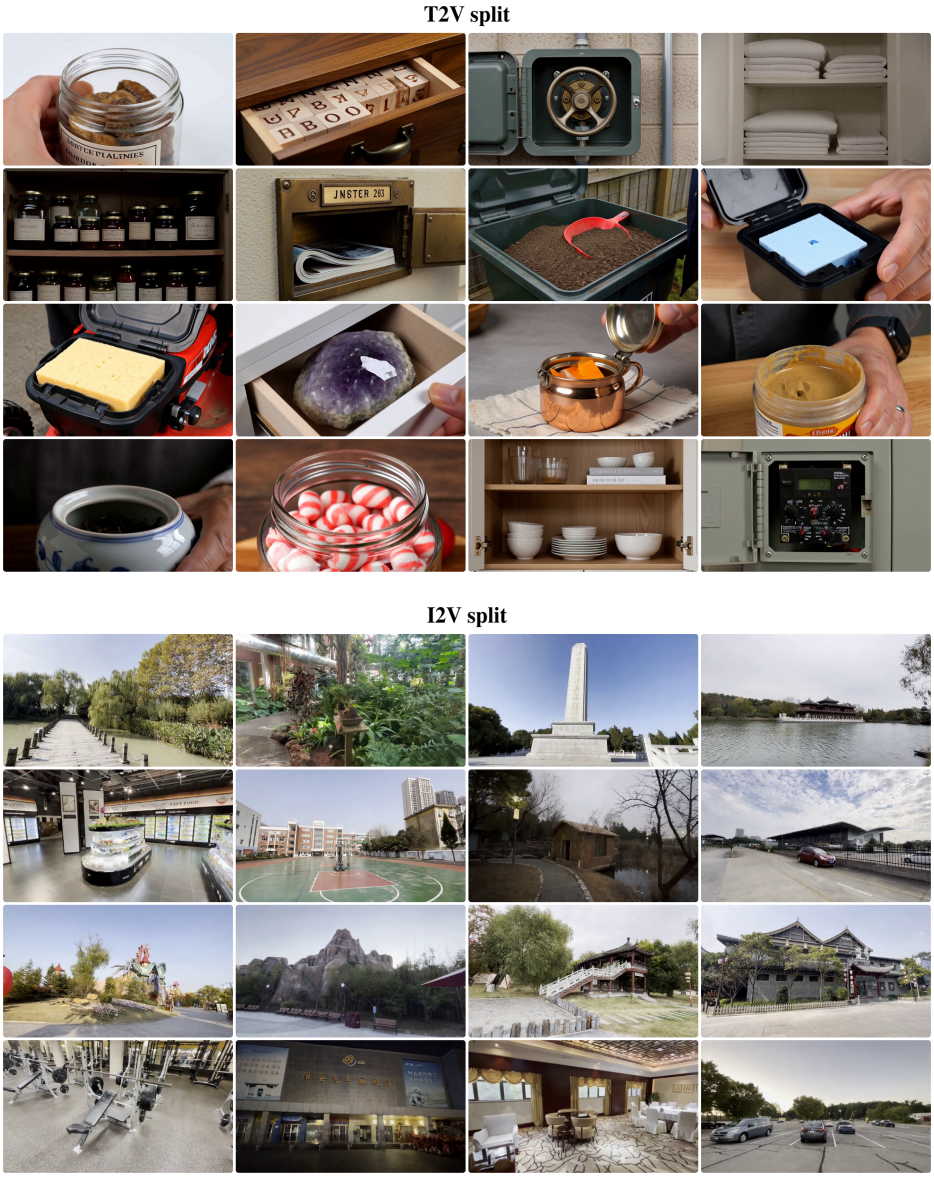

Figure 8: **Example frames from the two splits.** Top: model-generated frames from the T2V split when the prompt first reveals the object. Bottom: conditioning frames from the I2V split, taken from the first frames of DL3DV videos. Sixteen scenes are sampled at random from each split.

For rotation, the turnaround is the frame with the largest angular deviation of its viewing direction from that of the conditioning frame. For translation, it is the frame farthest from the initial position. Among frames from the turnaround onward, we select the revisit frame with the smallest unsigned angle between its reconstructed viewing direction and that of the conditioning frame. This rule also handles a full 360° turn without an angle-wrap ambiguity. CLIP and LPIPS are evaluated on the selected original images.

Metrics and aggregation. Table 9 specifies the metrics used for the T2V and I2V comparisons. CLIP and LPIPS compare the departure–revisit pair; TempSSIM and Drift summarize the full roll-out. For the latter two, we first average within each video, then report the median over scenes, as for the pairwise metrics. We use the same definitions for every method and memory budget.

Table 8: ***RememBench* evaluation settings.** The same inputs are used for all methods. Outdoor I2V scenes support both trajectory types.

	T2V	I2V
Scenes	100	50 indoor + 100 outdoor
Backbone	H3-AR	LingBot-World-Infinity
Resolution	1376×768	832×480
Frames / frame rate	379 / 24 fps	253 / 16 fps
Revisit control	Four-segment prompt	Camera trajectory
Rotation	—	90°, 180°, 360°; 150 scenes
Rotation + translation	—	Same angles; 100 outdoor scenes
Departure frame	Manually marked	Conditioning frame
Revisit frame	Manually marked per rollout	Selected from Pi3X poses

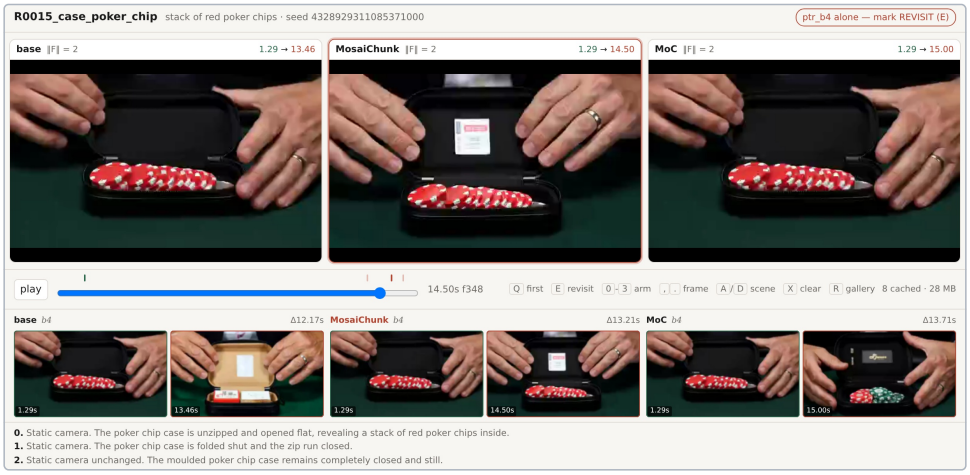

Figure 9: **T2V frame annotation.** During revisit annotation, the timeline controls only the selected rollout (red outline); the other videos remain paused. Q marks the shared departure frame, and E marks the selected rollout's revisit frame. Saved departure–revisit pairs appear below the timeline.

Table 9: **Metrics on *RememBench*.** CLIP and LPIPS assess revisit consistency; TempSSIM and Drift provide complementary measures of rollout quality. Arrows indicate the preferred direction.

Metric	Definition
CLIP ↑	Cosine similarity of L2-normalized image embeddings from CLIP ViT-H/14, using the LAION-2B checkpoint and its standard image processor.
LPIPS ↓	LPIPS with AlexNet (version 0.1), evaluated at each backbone's native image resolution after scaling pixels to $[-1, 1]$.
TempSSIM ↑	Mean SSIM over all consecutive decoded frame pairs in grayscale, with an 11×11 Gaussian window and $\sigma = 1.5$.
Drift ↓	Mean cosine distance between adjacent chunks. Each chunk is represented by the normalized average of CLIP embeddings from four evenly spaced frames within that chunk.

For Drift, chunk boundaries follow the backbone's decoded output: H3-AR has an initial five-frame chunk followed by 17-frame chunks; LingBot-World-Infinity has an initial 13-frame chunk followed by 16-frame chunks. This avoids treating an arbitrary fixed-length frame partition as the model's chunk structure.

C Additional Results

C.1 Extra Qualitative Results

Figures 10 and 11 show three scenes per setting, using the frame-pair protocol in Section B.3. The first and last columns show departure and revisit frames, with intermediate frames illustrating the intervening motion.

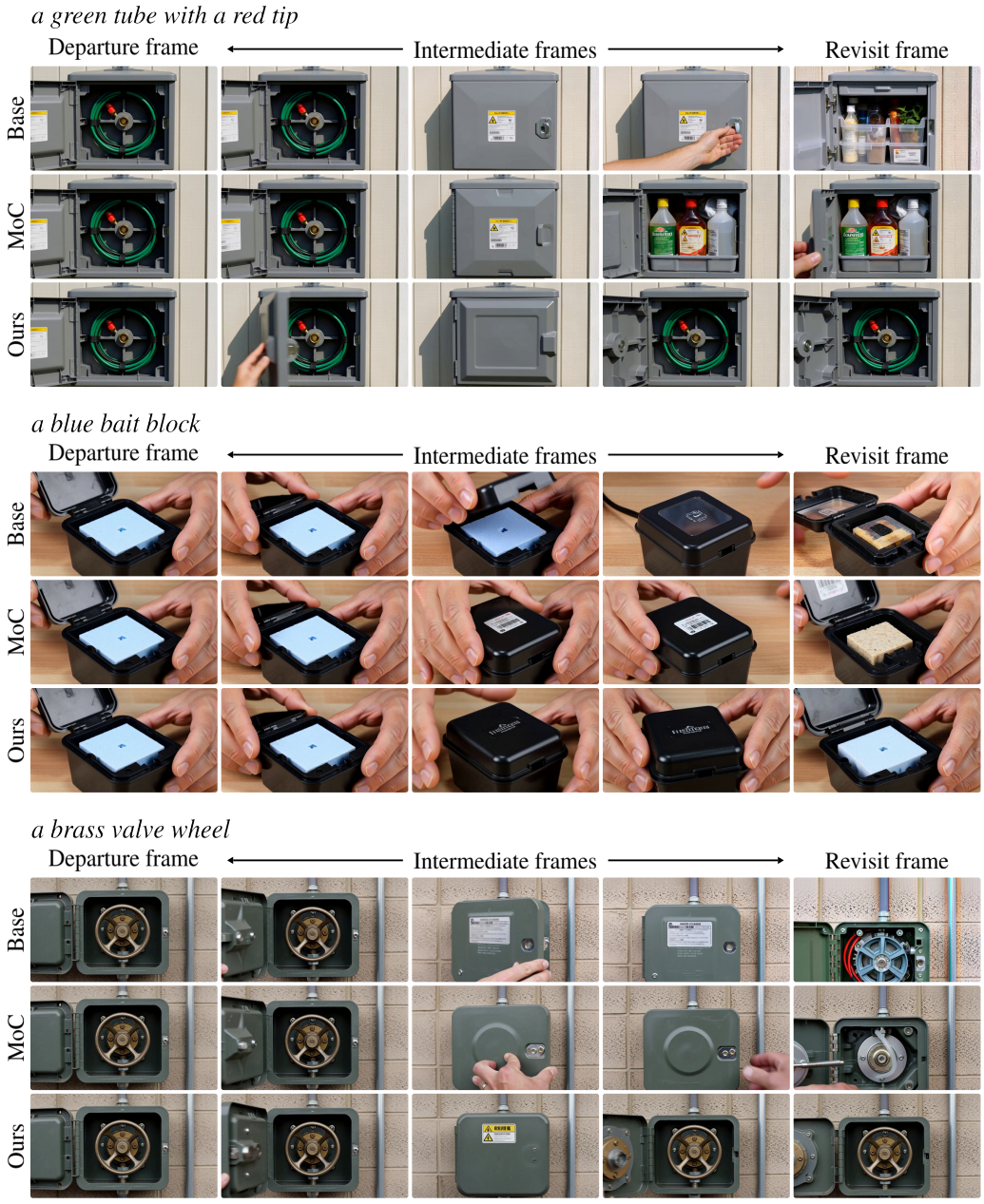

Figure 10: **Additional T2V revisits at a two-chunk far-memory budget.** The first column shows the departure frame before the target content leaves view; the last shows each method's marked revisit frame after it returns. Intermediate columns sample the rollout. Ours denotes *MosaiChunk*.

Within each scene, methods share the prompt and noise seed; I2V methods also share the conditioning frame and input camera trajectory. Rows therefore differ only in the memory supplied to the backbone. Rows are Base, MoC, and Ours (*MosaiChunk*), from top to bottom. The T2V examples

use two chunks of far memory; the I2V examples use one, as stated in the captions. The accompanying `video.html` provides synchronized full rollouts together with the departure–revisit frame pairs and their CLIP scores.

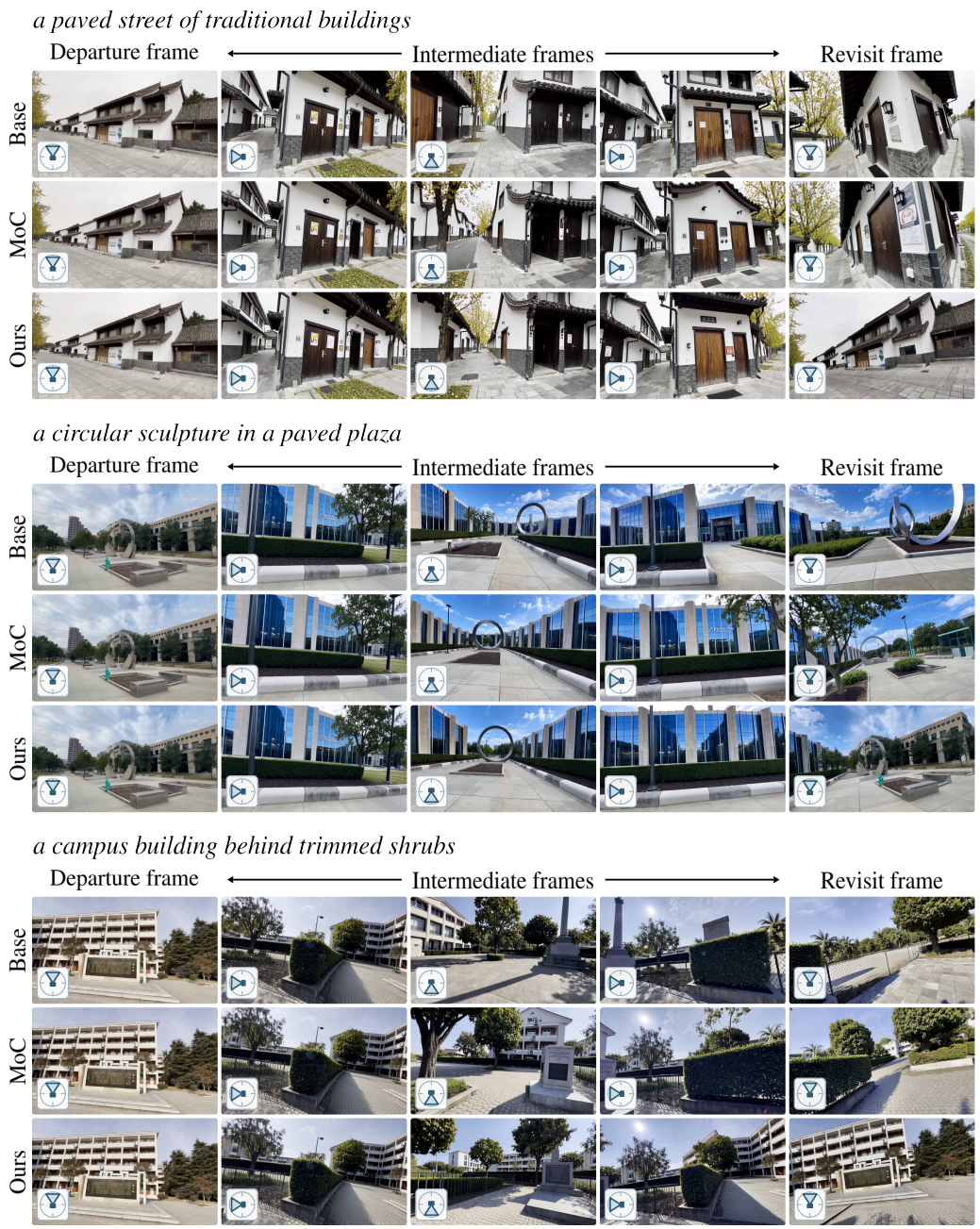

Figure 11: **Additional I2V revisits at a one-chunk far-memory budget.** All methods start from the same conditioning frame and follow the same 180° trajectory. The first column shows the conditioning frame, the intermediate columns sample the camera turn, and the last shows each method's selected revisit frame. Camera insets schematically show the commanded rotation. Ours denotes *MosaiChunk*.

C.2 Extra Quantitative Results

Public benchmarks. We evaluate I2V generation on WBench (Ying et al., 2026) and WorldMark (Xu et al., 2026), two general-purpose benchmarks for interactive video world models, and report every metric in their consistency and video-quality dimensions. We compare Base and *MosaiChunk* at one- and two-chunk far-memory budgets. Base matches the total active KV cache budget by retaining additional recent chunks instead of far memory. Scores are reported on a 0–100 scale, with higher values indicating better performance. Both benchmarks use the frozen LingBot-World-Infinity backbone at 832×480 and 16 fps, with four denoising steps per chunk and identical noise seeds across compared methods.

Table 3 summarizes selected metrics; full results and metric descriptions follow.

The relatively small camera rotations in these evaluations can preserve substantial overlap with earlier views, as in our 90° setting, reducing the need to recall content beyond the sliding window. The sliding-window baseline already achieves strong consistency scores in these evaluations. Nevertheless, *MosaiChunk* improves every consistency metric reported in Table 3 at both budgets, while also improving the video-quality scores.

WBench. We use the navigation split with 158 cases. Spatial and Gated Spatial are defined on its 60 round-trip cases, and Subject is evaluated on 103 cases; the remaining metrics use all 158 cases. The released adapter converts navigation commands into camera poses and combines segment prompts while preserving perspective information and command durations. Table 10 reports the full results. The metric descriptions below follow the released evaluation implementation.

For *consistency*, *Background* compares CLIP features across time. *Spatial* compares the initial and returning views using DreamSim, with the return frame selected from estimated camera orientations. *Gated Spatial* downweights this similarity when intermediate frames show little departure from the initial view. *Segment* reports the fraction of videos without detected shot boundaries. *Perspective* measures the stability and presence of a tracked subject's image-plane centroid. *Subject* compares masked subject features across adjacent frames and against the first frame. *Geometric* measures agreement between estimated depths after reprojection, and *Photometric* measures RGB agreement under the same geometric warping.

For *video quality*, *Aesthetic* uses a learned aesthetic predictor and *Imaging* uses MUSIQ to assess technical image quality. *Flickering* penalizes pixel changes between successive frames, *Dynamic* detects motion from optical flow, and *Smoothness* evaluates frame-interpolation error. *HPSv3-Norm* reports percentile-normalized human-preference scores.

Table 10 shows higher Background, Spatial, Gated Spatial, Perspective, Subject, and Geometric scores for *MosaiChunk* at both budgets. The improvements are not uniform across all metrics: Segment is lower at two chunks, and Photometric is slightly lower at that budget.

WorldMark. We evaluate the Real and Stylized first-person splits, each containing 125 videos per method and budget, and macro-average the scores across the two splits. Table 11 gives the complete results for the World Memory and Visual Quality dimensions. We use the supplied conditioning frames, prompts, and image-specific intrinsics, adjusted after center cropping. Each action segment lasts 20 seconds and commands 120° of rotation or two world units of translation. Completing the final generation chunk produces 333, 653, or 973 frames for one-, two-, or three-segment trajectories, respectively.

Local Memory penalizes detected cuts and abrupt changes in DINOv2 features between frames sampled at equal increments of accumulated motion. *Global Memory* measures geometric consistency across the trajectory using VGGT-Omega reconstruction. *Revisit Memory* aligns outward and return frames by cumulative optical-flow arc length and compares DINOv2 features of the matched pairs, using clips that satisfy the benchmark's return-motion checks. *Aesthetic* and *Perceptual* use the aesthetics and visual-quality tasks of Q-Align, respectively.

Table 11 shows improved Global and Revisit Memory at both budgets, alongside higher Aesthetic and Perceptual scores. Local Memory is slightly lower, distinguishing improved recall from local temporal stability.

Table 10: **Full WBench results.** Consistency and Video Quality metrics on the navigation split: Spatial and Gated Spatial use 60 round-trip cases, Subject uses 103 cases, and the remaining metrics use all 158 cases. Budgets specify *MosaiChunk*'s far memory in chunk equivalents; Base matches the total active KV cache budget by retaining additional recent chunks. Higher scores are better. The higher consistency score at each budget is bold, except ties; video-quality scores are not bolded.

	1 chunk		2 chunks	
Metric	Base	*MosaiChunk*	Base	*MosaiChunk*
Consistency				
Background	91.33	**91.57**	91.44	**91.52**
Spatial	78.28	**79.63**	79.43	**80.73**
Gated Spatial	75.47	**76.69**	76.78	**78.18**
Segment	97.47	97.47	**97.47**	94.94
Perspective	80.21	**82.11**	80.59	**82.14**
Subject	88.13	**88.36**	88.10	**88.38**
Geometric	85.98	**86.74**	86.14	**87.08**
Photometric	79.62	**79.67**	**79.85**	79.83
Video Quality				
Aesthetic	61.75	61.86	61.76	61.92
Imaging	67.31	67.42	67.40	67.45
Flickering	91.32	91.30	91.37	91.28
Dynamic	94.94	95.57	96.20	96.20
Smoothness	96.40	96.44	96.45	96.42
HPSv3-Norm	70.31	70.42	70.47	70.70

Table 11: **Full WorldMark results.** All World Memory and Visual Quality metrics, macro-averaged over the Real and Stylized first-person splits, with 125 videos per method and budget in each split. Budgets specify *MosaiChunk*'s far memory in chunk equivalents; Base matches the total active KV cache budget by retaining additional recent chunks. Higher scores are better. The higher consistency score at each budget is bold; video-quality scores are not bolded.

	1 chunk		2 chunks	
Metric	Base	*MosaiChunk*	Base	*MosaiChunk*
Consistency				
Local Memory	**95.55**	95.29	**93.70**	93.46
Global Memory	62.15	**62.22**	61.75	**63.00**
Revisit Memory	80.14	**84.89**	84.63	**84.87**
Video Quality				
Aesthetic	61.43	62.48	61.05	62.89
Perceptual	85.49	86.45	84.97	86.64

Additional camera trajectories. Table 12 reports 90° and 360° turns, with and without translation, using the trajectories in Section B.1 and frame-pair selection in Section B.3. At 90°, the starting view remains substantially visible and the methods achieve similar scores. At 360°, revisit consistency is lower for all methods and their relative ordering varies across settings. Thus, the benefits observed for the 180° revisit task do not extend uniformly to every camera trajectory: limited turns preserve overlapping visual context, while full rotations remain challenging.

Table 12: **I2V results at 90° and 360°.** Median revisit CLIP and LPIPS on 150 rotation scenes and 100 scenes with rotation and translation. Retrieval methods use one- or two-chunk far-memory budgets with a fixed sliding window; Base matches the total active KV cache budget by retaining additional recent chunks. WorldKV and *MosaiChunk-Pose* use camera poses; MoC scores pooled keys, while *MosaiChunk* uses learned section descriptors.

				CLIP ↑		
Trajectory	Budget	Base	MoC	*MosaiChunk*	WorldKV	*MosaiChunk-Pose*
90°	1 chunk	0.910	0.910	0.913	0.915	0.912
	2 chunks	0.907	0.908	0.906	0.915	0.910
90° + translation	1 chunk	0.882	0.884	0.889	0.900	0.896
	2 chunks	0.881	0.875	0.887	0.895	0.889
360°	1 chunk	0.716	0.722	0.717	0.728	0.704
	2 chunks	0.711	0.719	0.719	0.714	0.717
360° + translation	1 chunk	0.728	0.739	0.730	0.758	0.727
	2 chunks	0.733	0.726	0.752	0.730	0.748
				LPIPS ↓		
Trajectory	Budget	Base	MoC	*MosaiChunk*	WorldKV	*MosaiChunk-Pose*
90°	1 chunk	0.530	0.536	0.538	0.526	0.529
	2 chunks	0.554	0.551	0.551	0.529	0.542
90° + translation	1 chunk	0.564	0.571	0.563	0.529	0.530
	2 chunks	0.558	0.585	0.573	0.544	0.553
360°	1 chunk	0.691	0.688	0.694	0.693	0.686
	2 chunks	0.698	0.687	0.691	0.690	0.693
360° + translation	1 chunk	0.693	0.676	0.682	0.683	0.668
	2 chunks	0.681	0.673	0.677	0.673	0.675

C.3 Extra Ablations

Per-query selection details. For the per-query selection ablation, we retrain the descriptor encoder to select historical sections independently for each query section in the latest chunk. Selecting one or two sections per query gives the one- or two-chunk far-memory budget, respectively.

Camera pose as a retrieval signal. We examine whether camera poses improve descriptor-based retrieval and whether section composition remains useful when this guidance is available. *MosaiChunk* scores historical sections using learned descriptor similarity. *MosaiChunk-Pose* first uses camera-pose similarity to shortlist historical chunks, then applies the same trained router to select sections within them. For a whole-chunk comparison, we implement WorldKV's pose-based retrieval rule (Yi et al., 2026) on the same frozen backbone. Table 13 evaluates 180° trajectories on 150 rotation scenes and 100 scenes with rotation and translation, at one- and two-chunk far-memory budgets.

Pose guidance improves both CLIP and LPIPS over *MosaiChunk* in every setting. Relative to WorldKV, *MosaiChunk-Pose* achieves similar CLIP, with slightly higher scores at both two-chunk budgets. WorldKV is stronger on rotation at one chunk and has lower LPIPS under rotation with translation. Figure 12 illustrates how section composition can preserve the staircase of a background building, even though it occupies only a small part of the frame.

Far-memory budget. We compare allocating additional KV entries to retrieved far memory with retaining additional recent chunks. Retrieval methods keep their sliding window fixed, whereas Base matches the total active KV cache budget by retaining additional recent chunks. On 100 T2V and 150 I2V rotation scenes, we evaluate *MosaiChunk* and, for I2V, *MosaiChunk-Pose* at far-memory budgets from 0.5 to 2 chunks in half-chunk increments. The other methods are evaluated at one and two chunks (Figure 13).

Table 13: **Effect of pose-guided retrieval.** Revisit CLIP and LPIPS (medians) and rollout-quality metrics on 150 rotation scenes and 100 scenes with rotation and translation, all with 180° turns. WorldKV and *MosaiChunk-Pose* use camera poses; *MosaiChunk* uses learned section descriptors without pose guidance. Far-memory budgets are measured in original video-chunk equivalents, with the sliding window fixed. Better CLIP and LPIPS among the two pose-guided methods are bold.

Method	CLIP ↑	LPIPS ↓	TempSSIM ↑	Drift ↓
Far memory: 1 chunk				
Rotation				
MosaiChunk	0.840	0.625	0.456	0.051
WorldKV	**0.853**	**0.610**	0.458	0.051
MosaiChunk-Pose	0.843	0.624	0.459	0.051
Rotation + translation				
MosaiChunk	0.848	0.606	0.519	0.044
WorldKV	0.880	**0.543**	0.518	0.043
MosaiChunk-Pose	**0.881**	0.573	0.520	0.045
Far memory: 2 chunks				
Rotation				
MosaiChunk	0.863	0.609	0.458	0.051
WorldKV	0.891	0.578	0.461	0.050
MosaiChunk-Pose	**0.893**	**0.577**	0.461	0.050
Rotation + translation				
MosaiChunk	0.862	0.584	0.513	0.046
WorldKV	0.897	**0.542**	0.519	0.045
MosaiChunk-Pose	**0.899**	0.546	0.514	0.047

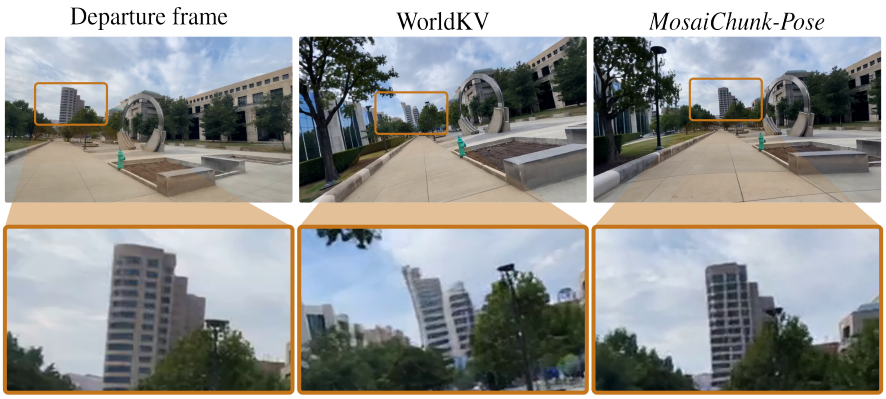

Figure 12: **Pose-guided retrieval at a one-chunk far-memory budget.** The left panel is the conditioning frame. Compared with WorldKV, *MosaiChunk-Pose* better preserves the background building's staircase.

Both *MosaiChunk* and *MosaiChunk-Pose* improve as the budget grows. The advantage also holds with less far memory: on T2V, *MosaiChunk* with one chunk exceeds MoC with two (0.899 vs. 0.841 CLIP); on I2V, half a chunk already does so (0.815 vs. 0.807). These comparisons demonstrate both effective retrieval at small budgets and improved revisit consistency when more memory is available.

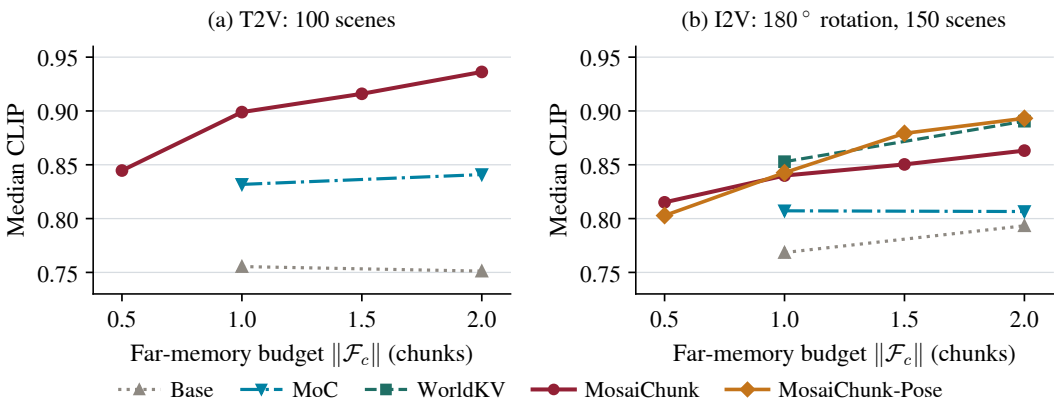

Figure 13: **Revisit consistency across far-memory budgets.** Median CLIP on 100 T2V scenes and 150 I2V 180° rotation scenes. *MosaiChunk* and *MosaiChunk-Pose* use half-chunk increments; other methods use one- and two-chunk budgets. Base matches the total active KV cache budget by retaining additional recent chunks.

C.4 KV Transplantation

We test whether selected historical KV can transfer content between rollouts without updating the video generator. We generate two cookie-tin rollouts with different cookies, manually mark each cookie in an earlier frame, and select the cached KV at the corresponding latent positions. When the tins close and reopen, we exchange these entries as far memory while keeping the generator frozen. Figure 14 shows that cookie A appears in tin B and cookie B appears in tin A, while each rollout retains its original tin. This intervention provides further evidence that historical KV can serve as content-specific memory even when it comes from another rollout.

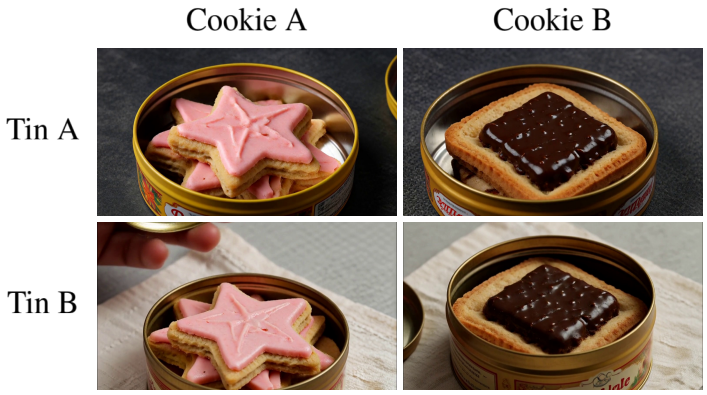

Figure 14: **Transplanting content between rollouts.** Rows specify the tin and columns the cookie. Diagonal images are source frames; off-diagonal images show the swapped results.