A^{2}RD: AGENTIC AUTOREGRESSIVE DIFFUSION FOR LONG VIDEO CONSISTENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

While diffusion models synthesize high-fidelity short clips, transforming them into consistent and coherent long videos remains challenging. Existing agentic approaches plan and coordinate segment generation, but often leave each segment unchanged once generated, leading to persistent semantic drift over long horizons. We introduce Agentic Auto-Regressive Diffusion (**A^{2}RD**), a training-free agentic architecture that decouples creative synthesis from consistency enforcement. A^{2}RD formulates long video synthesis as a closed-loop autoregressive process that synthesizes and self-improves video segment-by-segment against the evolving video world through a *Retrieve–Synthesize–Refine–Update* cycle. Its architecture combines *(i) Multimodal Video Memory*, which tracks visual and narrative states across segments; *(ii) Adaptive Segment Generation*, which selects extrapolation or interpolation according to the narrative transition; and *(iii) Hierarchical Test-Time Self-Improvement*, which refines boundary frames and video segments at their respective levels. We further introduce **LVBENCH-C**, a challenging benchmark with non-linear entity and environment transitions to stress-test long-horizon consistency. Across public and LVBENCH-C benchmarks spanning one- to ten-minute videos, A^{2}RD outperforms state-of-the-art baselines by up to 30% in consistency and 20% in narrative coherence. Human evaluations corroborate these gains while also highlighting notable improvements in motion and transition smoothness.

1 INTRODUCTION

Video synthesis has emerged as a transformative capability in artificial intelligence, powering high-impact applications including cinematic storytelling, educational content, entertainment, and advertising (Elmoghany et al., 2025; Ma et al., 2025). Although recent breakthroughs in diffusion models (Ho et al., 2022; Singer et al., 2023; Esser et al., 2023; Brooks et al., 2024; Wan et al., 2025a; Google Deepmind, 2025; ByteDance Seed, 2026) have achieved remarkable fidelity for second-long clips, real-world applications demand minute- to hour-long videos. Scaling to coherent long video synthesis, however, remains a fundamental challenge. At its core are two fundamental problems: *temporal consistency*, which requires models to track and preserve entities, environments, and motion dynamics, and *narrative coherence*, which demands that videos evolve meaningfully over time.

State-of-the-art long-video synthesis approaches have achieved remarkable progress in addressing these challenges. However, they mostly follow the passive, open-loop paradigm, which has limitations. *Frame-based autoregressive (FAR)* models synthesize videos frame-by-frame or chunk-by-chunk (Huang et al., 2025a; Yang et al., 2026a; Chen et al., 2026), naturally preserving local temporal continuity. However, these models employ generated frames as fixed conditioning for subsequent generation, causing errors to propagate uncorrected and limiting narrative controllability. This often leads to semantic drift and narrative repetition over long horizons (Zhao et al., 2026). Meanwhile, existing *segment-based* methods synthesize and concatenate short segments, either autoregressively (SAR) (Zhou et al., 2026; An et al., 2026; Zhang et al., 2025a) or in parallel (Wu et al., 2025c; Wang et al., 2026), offering stronger narrative control. However, they struggle to maintain inter-segment consistency and continuity (Elmoghany et al., 2025). While recent SAR methods use memory to maintain visual references across segments, they typically do not verify and repair a generated segment before it becomes context for subsequent generation. Errors can therefore persist and propagate across segments (Figures 5 and 6 and Section I).

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.

To address these limitations, we develop A^{2}RD (*/a:rd/*, Agentic Auto-Regressive Diffusion), a closed-loop, agentic architecture for autoregressive long-video synthesis. By integrating segment-level test-time self-improvement into autoregressive generation, A^{2}RD self-improves each segment for consistency, coherence, and quality against the evolving video world before its output becomes context for subsequent generation. See Figure 4 for examples. A^{2}RD is training-free and built upon three pillars: a Multimodal Video Memory, an Adaptive Segment Generation, and Hierarchical Test-Time Self-Improvement algorithms. Together, these components form a Retrieve–Synthesize–Refine–Update cycle: for each segment, the agent retrieves relevant video world contexts from memory, determines the segment generation mode (e.g., extrapolation or interpolation) adaptively, synthesizes boundary frames then video segment with hierarchical self-improvements applied at both frame and video levels, and finally updates the memory for following generation (Figure 1).

Moreover, existing benchmarks lack the realistic complexity of long-horizon narratives, where entities and environments undergo non-linear transitions. We contribute LVBENCH-C, a benchmark designed to challenge long-horizon consistency where entities and environments appear, disappear, and reappear (*"cyclic"*) with optional state changes. Extensive evaluations on LVBENCH-C and public benchmarks show that A^{2}RD achieves state-of-the-art consistency and narrative coherence in just two self-improved iterations, corroborated by human studies. In summary, we contribute:

- We introduce A^{2}RD, a closed-loop agentic autoregressive architecture for long-video synthesis that addresses the cross-segment consistency and narrative coherence challenges through multimodal memory, adaptive generation, and hierarchical self-improvement.
- We introduce LVBENCH-C, a challenging benchmark stress-testing long-horizon video consistency through cyclical entity and environment appearances with optional state evolutions.
- We conduct extensive experiments, including compute-matched comparisons, fully open-weight model stacks, component ablations, and human evaluations, demonstrating state-of-the-art consistency and narrative coherence on one- to ten-minute videos.

2 Related Work

Long-Form Video Synthesis. State-of-the-art (SOTA) long video synthesis approaches follow two main paradigms. Frame-based autoregressive models generate frames or chunks autoregressively, conditioning each on prior content via rolling KV caches (Huang et al., 2025a), short-window attention with frame-level sinks (Yang et al., 2026a), or initial-frame anchoring (Liu et al., 2026). While preserving local visual fidelity, they remain prone to semantic drift and content repetition, and offer limited controllability (Zhao et al., 2026). Segment-based methods synthesize segments in parallel, with (Wang et al., 2025a; Meng et al., 2026) or without shared denoising (Yin et al., 2023; Wu et al., 2025c; Wang et al., 2026), or autoregressively (SAR) (An et al., 2026; Zhang et al., 2025a; Zhou et al., 2026). These offer finer narrative control but struggle with inter-segment consistency (Elmoghany et al., 2025). Segments are typically synthesized via *extrapolation* from a begin frame (Wang et al., 2026; An et al., 2026; Zhou et al., 2026) or *interpolation* between planned boundaries (Yin et al., 2023). Yet, each has limitations: extrapolation often causes inconsistencies for details absent from the begin frame, while interpolation can yield unnatural progression from poorly planned boundary frames. Existing methods also lack mechanisms to correct such errors, causing them to propagate across segments (Figure 6, Section I). A^{2}RD addresses these limitations by coupling SAR with an adaptive generation strategy, multimodal memory for richer conditioning, and closed-loop self-improvement, achieving strong temporal consistency, and narrative controllability.

Test-Time Scaling for Generative Models. Test-time scaling (TTS) improves generation quality by investing additional computation during inference (Snell et al., 2025). For image, this includes best-of-N sampling (Zhang et al., 2025b), iterative refinement (Zhuo et al., 2025; Qu et al., 2026), prompt optimization (Wan et al., 2025b; Wang et al., 2024a; Mañas et al., 2024), and evolutionary search (He et al., 2025). Video TTS methods have recently emerged (Gao et al., 2025; Long et al., 2026; Huang et al., 2025b; Zhu et al., 2026; Yang et al., 2026b; Hong et al., 2026), primarily focusing on prompt optimization: RAPO (Gao et al., 2025) enriches prompts through retrieval-augmented refinement, VISTA (Long et al., 2026) employs multi-agent iterative planning and critique, and VQQA (Song et al., 2026) uses VLM-generated questions for closed-loop optimization. However, these methods optimize individual clips without evaluating them against the evolving entity, environment, and narrative states of a long video. A^{2}RD integrates test-time self-improvement into autoregressive

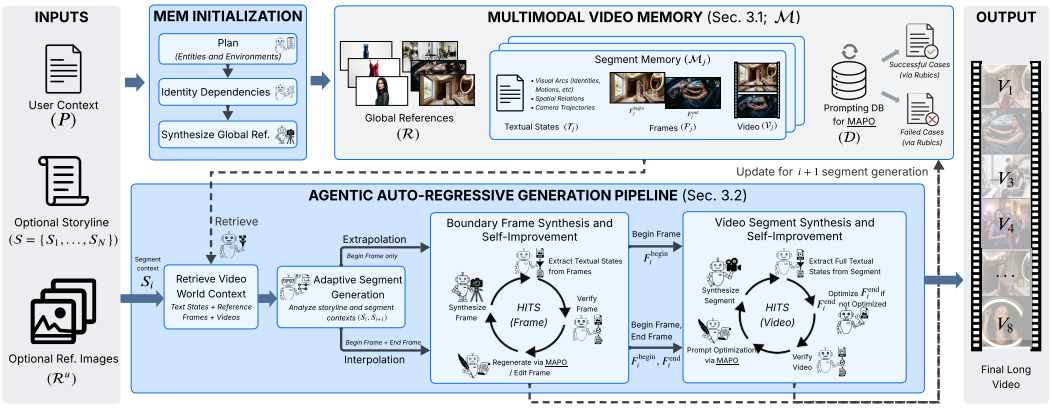


Figure 1: Overview of A^{2}RD. For each segment, it retrieves relevant memory, selects extrapolation or interpolation, and hierarchically self-improves the boundary frames and video segment.

generation, improving boundary frames and video segments against cross-segment context to enforce consistency and narrative coherence.

3 A^{2}RD: Agentic Auto-Regressive Diffusion

Figure 1 outlines on overview of A^{2}RD. We term our basic generation unit as **"segment"** (equivalent to "clip"), a flexible unit that can span one or more scenes or shots. A^{2}RD takes as input a user context P, a storyline $\mathcal{S} = \{S_1, \ldots, S_N\}$ (provided or planned from P) with S_i being i-th segment context, and optional reference images $\mathcal{R}^u$. The agent supports any Text-Image-to-Video (TI2V) model via incorporating a Multimodal Large Language Model (MLLM) and a Text-Image-to-Image (TI2I) model. It begins by initializing a Multimodal Video MEMory (MVMem) via synthesizing global entity and environment references $\mathcal{R}$, then synthesizing video segments autoregressively, continuously retrieving and updating the MVMem for context-aware synthesis and self-improvement.

3.1 MVMem Design and Initialization

MVMem enables A^{2}RD to explicitly track evolving video world states and events, thus enforcing long-range dependencies for temporal consistency and narrative coherence across segments.

Memory Schema. Unlike existing studies that focus extensively on visual references (Section C.4), MVMem stores structured contexts from synthesized segments, denoted as $\mathcal{M} := \{\mathcal{M}_1, \ldots, \mathcal{M}_N\} \cup \mathcal{R} \cup \mathcal{D}$. Here, $\mathcal{R}$ denotes global reference frames (including $\mathcal{R}^u$), and $\mathcal{D}$ is the prompt database. Each segment memory $\mathcal{M}_j := \{T_j, \mathcal{F}_j, V_j\}$ disentangles the video segment into three modalities:

(i) Textual States (T_j). To capture the evolving narrative for consistency and coherence, we model the video's underlying state as a structured, fine-grained representation, inspired by Johnson et al. (2015). T_j consists of: *(1) Visual Arcs* that track entity and environment features and their temporal evolution, recording elements' Identity, Identity Changes, and Motion; *(2) Spatial Relations* that capture subject-relation-object triplets from the begin frame to ground geometric layouts; and *(3) Camera* states that record viewport trajectories for visual continuity. We extract T_j hierarchically: first deriving elements' Identity and Spatial Relations from the begin frame (T_j^F), then supplementing missing elements, Identity Changes, Motions, and Camera dynamics from the full segment to form T_j. This decouples frame-level from video-level extraction for A^{2}RD's pipeline.

(ii) Frames ($\mathcal{F}_j$ or $\mathcal{R}$). To anchor the concrete visual details that text cannot fully articulate, MVMem stores global reference frames $\mathcal{R}$ (both synthesized and user-provided, $\mathcal{R}^u \subseteq \mathcal{R}$), each indexed by a generated caption, and segment keyframes $\mathcal{F}_j := \{F_j^{\text{begin}}, F_j^{\text{end}}\}$, indexed by S_j. Our framework can accommodate more advanced frame extraction and indexing methods.

(iii) Videos (V_j). To capture temporal motion dynamics for cross-segment smooth transitions and motion continuity, MVMem saves the synthesized segments for segment verification and refinement. Like keyframes, V_j is simply indexed by S_j.

MVMEM enables two core online operations: *Retrieve* fetches relevant past states and *Update* writes the newly synthesized $\mathcal{M}_j$ for subsequent generation, see below. $\mathcal{D}$ is described in Section 3.3.

MVMEM Initialization. Before synthesizing segments, inspired by identity-reference approaches (Zheng et al., 2024; Liu et al., 2026) for consistency, A^{2}RD initializes MVMEM by establishing global reference backgrounds and entities, $\mathcal{R} := \mathcal{R}^{\text{bg}} \cup \mathcal{R}^{\text{ent}} \cup \mathcal{R}^{u}$ as a form of long-term memory:

(i) Planning. The agent first reasons over $\mathcal{S}$ (and $\mathcal{R}^{\text{u}}$ if available) to identify the environments and entities, their required appearances (both explicitly specified and implicitly implied from $\mathcal{S}$), and generates prompts for synthesizing these references, using the MLLM:

$$\mathcal{P}^{\text{bg}}_{\mathcal{R}} := \text{MLLM}^{\text{bg}}_{\text{plan}}(P, \mathcal{S}, \mathcal{R}^{u}), \quad \mathcal{P}^{\text{ent}}_{\mathcal{R}} := \text{MLLM}^{\text{ent}}_{\text{plan}}(P, \mathcal{S}, \mathcal{R}^{u}), \quad \mathcal{P}^{u}_{\mathcal{R}} := \text{MLLM}_{\text{cap}}(P, \mathcal{S}, \mathcal{R}^{u}) \tag{1}$$

where $\mathcal{P}_{\mathcal{R}} := \mathcal{P}^{\text{bg}}_{\mathcal{R}} \cup \mathcal{P}^{\text{ent}}_{\mathcal{R}} \cup \mathcal{P}^{\text{u}}_{\mathcal{R}}$ represents identified entities and environments' prompts, and $\mathcal{P}^{\text{u}}_{\mathcal{R}}$ denotes the captions of user-provided reference frames $\mathcal{R}^{\text{u}}$.

(ii) Identifying Dependencies. The agent constructs a dependency Directed Acyclic Graph $\mathcal{G}$ over $\mathcal{P}_{\mathcal{R}}$: $\mathcal{G} := \text{MLLM}_{\text{dep}}(\mathcal{P}_{\mathcal{R}})$, to identify which references depend on others (e.g., an entity depends on its environment). $\mathcal{G}$ is then decomposed into weakly connected components. Within each component, topological sorting is applied to determine the synthesis order to respect the dependencies.

(iii) Synthesizing References. The agent synthesizes a reference frame for each prompt in $\mathcal{P}_{\mathcal{R}} \setminus \mathcal{P}^{\text{u}}_{\mathcal{R}}$ using the TI2I, conditioned on its dependent references following topological order. All components are synthesized in parallel, yielding $\mathcal{R}$. See Section H.1 for our prompts.

3.2 AGENTIC AUTO-REGRESSIVE GENERATION PIPELINE

After establishing global references, A^{2}RD synthesizes long videos autoregressively, segment-by-segment. For each segment context S_i, the agent first determines the generation mode, then operates in a **Retrieve–Synthesize–Refine–Update** closed-loop, where Synthesize–Refine is applied first to boundary frames and then to the video segment. For convenience, we duplicate $S_{N+1} := S_N$ in $\mathcal{S} := \{S_1, \ldots, S_{N+1}\}$, and synthesize segments for $i \leq N$. See Section H.2 for our prompts.

(i) Adaptive Segment Generation. A key challenge for segment-based generation is balancing narrative progression with consistency. Prior works adopt either extrapolation (An et al., 2026; Zhou et al., 2026) or interpolation (Yin et al., 2023). Extrapolation allows natural video world progression but risks semantic drift, particularly for entities and environments not visible in the begin frame. Interpolation enforces stronger consistency, but risks unnatural progression especially when TI2I models lack the temporal reasoning to reliably synthesize how environments evolve over a predefined duration, given static references (Figure 5). A^{2}RD instead adaptively selects the mode per segment:

$$\text{mode}_i := \begin{cases} \text{extrapolation} & \text{if } \mathcal{C}(S_i) \wedge S_{i+1} \text{ is spatio-temporally continuous with } S_i \\ \text{interpolation} & \text{otherwise} \end{cases} \tag{2}$$

where $\mathcal{C}(S_i)$ indicates that S_i does not transition to a new established environment, and both conditions are inferred by $\text{MLLM}_{\text{mode}}(\mathcal{S})$. Interpolation applies when S_i is a multi-shot context whose shots span different environments, or when S_{i+1} jumps to a new environment. The second condition is omitted for $i = N$. See Figure 8 for an example of our mode selection.

(ii) Retrieve. After determining the mode, A^{2}RD retrieves text, image, and video contexts for synthesis. For any j-th segment context, to mitigate false-positive conditioning, the agent employs an MLLM to identify the top-k most narratively relevant previous segments. It acquires the textual states $\mathcal{T}^{\text{rel}}_j$, visual references $\mathcal{F}^{\text{rel}}_j$, and the immediate temporally contiguous segment $\mathcal{V}^{\text{rel}}_j$ (if any):

$$\mathcal{S}^{\text{rel}}_j := \text{MLLM}^{\text{scenes}}_{\text{retr}}(\mathcal{S}, k, S_j), \quad \mathcal{V}^{\text{rel}}_j := \text{MLLM}^{\text{cont}}_{\text{retr}}(\{V_{<j}\}, \mathcal{S}, S_j), \quad \mathcal{R}^{\text{rel}}_j := \text{MLLM}^{\text{ref}}_{\text{retr}}(\mathcal{R}, \mathcal{P}_{\mathcal{R}}, S_j) \tag{3}$$

$$\mathcal{T}^{\text{rel}}_j := \{T_n \mid S_n \in \mathcal{S}^{\text{rel}}_j\}, \quad \mathcal{F}^{\text{rel}}_j := \{F^{\text{begin}}_n, F^{\text{end}}_n \mid S_n \in \mathcal{S}^{\text{rel}}_j\} \cup \mathcal{R}^{\text{rel}}_j \tag{4}$$

When available, $\mathcal{F}^{\text{rel}}_j$ and $\mathcal{T}^{\text{rel}}_j$ are extended with F^{begin}_i and T^{F}_i respectively, ensuring subsequent synthesis for the current segment is conditioned on the begin frame's context.

(iii) Synthesize and Self-Improve–Boundary Frame(s). Based on Equation (2), A^{2}RD synthesizes boundary frames $\mathcal{F}_i := \{F^{\text{begin}}_i, F^{\text{end}}_i\}$. It assigns $F^{\text{begin}}_i := F^{\text{end}}_{i-1}$ for $i > 1$ for continuity. For

$i = 1$, it synthesizes F_1^{begin} via generating its frame prompt first, and then the frame $F_1^{\text{begin}} := \text{TI2I}(\text{MLLM}_{\text{pgen}}^{\text{img}}(S_1), \mathcal{F}_1^{\text{rel}})$. For $i > 1$, denote $\mathcal{V}_i^{\text{rel}} := \{V_t\}$, the end frame F_i^{end} is determined using the lookahead context of the subsequent segment i+1:

$$F_i^{\text{end}} := \begin{cases} \text{TI2I}(P_i^{\text{end}}, \mathcal{F}_{i+1}^{\text{rel}}) & \text{mode}_i = \text{interp},\ V_t \text{ does not exist} \vee t = i{-}1 \\ \text{MLLM}_{\text{retr}}^{\text{img}}(\mathcal{K}_t, S_t, S_{i+1}) & \text{mode}_i = \text{interp},\ V_t \text{ exists},\ t \neq i{-}1 \\ \emptyset & \text{mode}_i = \text{extrap} \end{cases} \tag{5}$$

where $P_i^{\text{end}} := \text{MLLM}_{\text{pgen}}^{\text{img}}(S_{i+1}, \mathcal{T}_{i+1}^{\text{rel}})$ is the generated frame prompt. The case $t \neq i{-}1$ is particularly challenging. It arises when segment i resumes some events from the middle of a non-adjacent segment t. To obtain the end frame of the relevant shot in segment t for resumption, we extract all shot end frames $\mathcal{K}_t$ from V_t (Castellano), then the MLLM selects the one that best continues into S_{i+1} (Section E.3 for an example). All synthesized frames by the TI2I model then undergo a frame-level self-improvement process, described in Section 3.3.

(iv) Synthesize and Self-Improve–Video Segment. After obtaining $\mathcal{F}_i$, A^{2}RD synthesizes the segment:

$$V_i := \text{TI2V}(P_i, \mathcal{F}_i, \mathcal{F}_i^{\text{rel}}) \tag{6}$$

where $P_i := \text{MLLM}_{\text{pgen}}^{\text{vid}}(S_i, \mathcal{T}_i^{\text{rel}}, \mathcal{F}_i)$ is the video segment prompt. Once synthesized, V_i undergoes a video-level self-improvement process, described in Section 3.3.

(v) Update. After self-improvements, MVMEM saves $(\mathcal{F}_i, V_i, T_i, T_{i+1}^F)$ if $i < N$ for reference in subsequent generation, where T_i and T_{i+1}^F are textual states extracted during refinement processes.

3.3 HIERARCHICAL BOUNDARY FRAME AND VIDEO SELF-IMPROVEMENT

To mitigate the risk of cascading temporal errors, where a single inconsistent frame can propagate artifacts across the entire horizon, A^{2}RD introduces **HIerarchical Test-time Self-improvement (HITS)** to self-improve synthesized frames and video segments hierarchically. Unlike existing works (Liu et al., 2025; Long et al., 2026) that apply search and closed-loop refinements to short clips, A^{2}RD extends this paradigm to self-improve both intra- and inter-segment coherence.

Boundary Frame Self-Improvement. This step self-improves F_i^* ($* \in \{\text{begin}, \text{end}\}$) interactively. At each iteration, A^{2}RD extracts frame textual states from the synthesized frame: T_i^F (Section 3.1) for the begin frame F_i^{begin}, and T_{i+1}^F for the optional end frame F_i^{end}, via $T_{i+\delta}^F := \text{MLLM}_{\text{ext}}^{\text{img}}(S_{i+\delta}, F_i^*)$ with $\delta \in \{0, 1\}$. $(F_i^*, T_{i+\delta}^F)$ is then verified via a 8-metric rubric focusing on consistency and basic image quality on a scale of 1–10 in 3 groups: (i) *Entity Consistency*, *Environment Consistency*, *Narrative Progression*, and *Spatial Logicalness*; (ii) *Entity State*, *Environment State*; (iii) *Instruction Following* and *Physical Plausibility*:

$$\mathcal{Q}^F(F_i^*) := \underbrace{\text{MLLM}_{\text{vfy}}^{\text{img-imgs}}(P_i^*, F_i^*, \mathcal{F}_{i+\delta}^{\text{rel}})}_{\text{(i) Inter Consistency over Images}} \cup \underbrace{\text{MLLM}_{\text{vfy}}^{\text{img-st}}(P_i^*, T_{i+\delta}^F, \mathcal{T}_{i+\delta}^{\text{rel}})}_{\text{(ii) Inter Consistency over States}} \cup \underbrace{\text{MLLM}_{\text{vfy}}^{\text{img-qual}}(S_{i+\delta}, P_i^*, F_i^*, T_{i+\delta}^F)}_{\text{(iii) Basic Quality}} \tag{7}$$

The agent then refines F_i^* via *Edit* or *Regenerate*: the outcomes of $\mathcal{Q}^F$ are input into the MLLM to decide the mode and, if *Edit* is chosen, to suggest the edit prompt. The Edit mode targets a single issue only, as it is challenging to fix multiple errors simultaneously. If Regenerate is chosen, P_i^* is optimized through our *Memory-Augmented Prompt Optimization (MAPO)* algorithm, see below, and F_i^* is re-synthesized for next iteration. The final F_i^* is:

$$F_i^* := \arg \max_{F \in \mathcal{F}_{\text{cand}}} \text{Agg}(\mathcal{Q}^F(F)) \tag{8}$$

where $\mathcal{F}_{\text{cand}}$ is the set of candidate frames generated across all refinement iterations.

Video Segment Self-Improvement. This step self-improves V_i interactively. Similar to above, A^{2}RD first extracts full video states: $T_i := \text{MLLM}_{\text{ext}}^{\text{vid}}(S_i, T_i^F, V_i)$ (Section 3.1). It then verifies (V_i, T_i) via a 10-metric rubric focusing on inter-segment consistency, intra-segment consistency, and basic video quality, divided into three groups, each scored on a scale of 1–10: (i) *Inter Entity Consistency*, *Inter Environment Consistency*, *Inter Motion Consistency*, *Camera Consistency*; (ii) *Intra Entity Consistency*, *Intra Environment Consistency*; and (iii) *Instruction Following*, *Physical Plausibility*, *Narrative Progression*, and *Frame Fit* (only when F_i^{end} is available):

$$\mathcal{Q}^V(V_i) := \underbrace{\text{MLLM}_{\text{vfy}}^{\text{vid-vid}}(\mathcal{S}_i^{\text{rel}}, S_i, S_t, T_i, T_t, V_i, V_t)}_{\text{(i) Inter Consistency}} \cup \underbrace{\text{MLLM}_{\text{vfy}}^{\text{vid-qual}}(S_i, P_i, V_i, T_i, F_i^{\text{end}})}_{\text{(ii), (iii): Intra Consistency and Basic Quality}} \tag{9}$$

The agent refines V_i depending on the availability of F_i^{end}. If F_i^{end} is available, the agent optimizes the text prompt P_i only. If F_i^{end} is unavailable, prompt-only optimization is insufficient, as entities and environments absent from F_i^{begin} or transformed during the segment can drift from references. In this case, A^{2}RD co-optimizes both F_i^{end} and P_i sequentially: it first extracts F_i^{end} from V_i, self-improves it following the frame self-improvement process above with *Edit* mode only (to preserve any natural layout progression), re-optimizes P_i via MAPO conditioned on the updated boundary frames $F_i := \{F_i^{\text{begin}}, F_i^{\text{end}}\}$, and then re-synthesizes V_i for the next iteration. The final V_i is:

$$V_i := \arg \max_{V \in \mathcal{V}_{\text{cand}}} \text{Agg}(\mathcal{Q}^V(V)) \tag{10}$$

where $\mathcal{V}_{\text{cand}}$ is the set of candidate videos generated across refinement iterations.

Memory-Augmented Prompt Optimization (MAPO). To improve the refinement efficacy, we introduce **MAPO**, which leverages the history of successful and failed cases indexed by rubric scores. Specifically, MVMEM maintains a prompt database $\mathcal{D} := \{(P, P^*, \mathcal{Q}, \ell)\}$ where each entry stores an original prompt P, its refined version P^*, rubric scores $\mathcal{Q}$, and a hard label $\ell \in \{\text{pos}, \text{neg}\}$ indicating positive and negative refinements. Each entry is indexed by a semantic embedding $\text{Emb}(P, \mathcal{Q})$ for efficient retrieval. $\mathcal{D}$ is seeded with a few prior cases and updated online: a case is assigned as "pos" if all rubric scores improve, and "neg" if all scores worsen. Given $\mathcal{Q}_i^F/\mathcal{Q}_i^V$ and the P_i^*/P_i, MAPO retrieves the top-k relevant positive and negative cases via cosine similarity over $(\mathcal{Q}, P)$ embeddings from $\mathcal{D}$. Inspire by Zhao et al. (2024), MAPO then contrasts the positive and negative cases to derive refinement guidelines, reasons over $\mathcal{Q}$ to identify root causes of failures, and applies targeted edits with guidelines to produce P^*. After each refinement, the new case is labeled and added to $\mathcal{D}$ online.

4 LONG VIDEO BENCH-CHALLENGE (LVBENCH-C)

Existing benchmarks (Huang et al., 2025c;c; Sun et al., 2025; Wu et al., 2025c; Zhang et al., 2025a) overlook real-world scenarios where entities and environments undergo natural non-linear transitions—appearing, disappearing, and reappearing with optional state changes across scenes (Section C.5 and Table 6). We introduce **LVBENCH-C**, a benchmark that stress-tests models' ability to maintain consistent and coherent world states in such scenarios, featuring three challenge types: (i) *Evolving Character States*, where characters reappear with evolved states (e.g., clothing, appearance, physical condition); (ii) *Evolving Object States*, where objects reappear with changed properties (e.g., position, condition); and (iii) *Evolving Environment States* (e.g., evolution, progressive revelation of details). We instantiate LVBENCH-C with 125 text-only scenarios: 50 samples each for 3- and 5-minute videos (30 character, 10 object, 10 environment), and 25 samples for 10-minute videos (10 character, 5 object, 5 environment). Each scenario consists of concise scene descriptions that either continue from previous scenes or transition to new ones, forming coherent narratives, see Figures 17 to 19.

We construct LVBENCH-C via a period-independent human-in-the-loop pipeline: for each challenge type, we craft a screenwriter persona prompt with one human-written demonstration and generate scenarios via a SOTA MLLM (Google DeepMind, 2025), enforcing constraints on content coherence, gap rules (entities absent for $\geq n{=}10$ segments before reappearing), and natural state changes, with manual review and prompt modification during the process to improve diversity. Raw scenarios are deduplicated by summarizing and comparing them using the same MLLM, followed by a customized Self-Refine (Madaan et al., 2023) pipeline where an MLLM-Judge validates each scenario against six criteria (specificity, logic, naturalness, realism, non-repetition, and consistency), refining failures iteratively. A separate SOTA MLLM (Anthropic, 2025) re-verifies and performs minimal refinements to avoid self-preference bias, with manual spot-checks confirming final quality, see Section F.3.

5 EXPERIMENTS

5.1 SETTINGS

Benchmarks. We conduct experiments on both single-scene and multi-scene long video generation. For single-scene, following Yang et al. (2026a); Yi et al. (2025), we use *VBench-Long* (Huang et al., 2025c) with 40 prompts, each decomposed into 8 continuous segments using Gemini (Google DeepMind, 2025), yielding approximately 1-min videos. For multi-scene, we use our LVBENCH-C

Method	Type	Semantic Alignment	Narrative Coherence	Inter-shot Consistency			Intra-shot Consistency	
				Character	Environment	Motions	Subject	Environment
Direct Prompting	Par	0.2012	0.3875	0.2695	0.3115	0.9710	0.7713	0.8929
Naive-AR	AR	0.1828	0.7466	0.5125	0.7148	0.9903	0.9062	0.9363
Naive-Par	Par	0.1814	0.6796	0.5500	0.6779	0.9836	0.8852	0.9148
MovieAgent (Wu et al., 2025c)	Par	0.1783	0.6071	0.5002	0.4944	0.9723	0.9150	0.9386
ViMax (HKUDS, 2025)	Par	0.1961	0.6912	0.5552	0.7015	0.9754	0.9215	0.9488
VideoMemory (Zhou et al., 2026)	AR	0.1926	0.6717	0.5729	0.7273	0.9777	0.9207	0.9492
A^{2}RD-Par (Ours)	Par	0.2094	0.8082	0.6817	0.8180	0.9843	0.9108	**0.9578**
A^{2}RD (Ours)	AR	**0.2111**	**0.8987**	**0.7353**	**0.8368**	**0.9935**	**0.9231**	0.9514

Table 1: Experiments on VBench-Long (Huang et al., 2025c) expanded to 8 continuous scenes for 1-min videos.

benchmark featuring challenging environment transitions and entity evolutions across 3-min videos (24 scenes) and 5-min videos (40 scenes), with 20 samples each, and 10-min videos in Section D.4.

Models and Baselines. We instantiate A^{2}RD and baselines with Gemini 3 Flash (Google DeepMind, 2025) as the MLLM, Nano Banana 2 (Raisinghani, 2026) as the TI2I, and Veo 3.1 (Google Deepmind, 2025) as the TI2V models. We also introduce A^{2}RD-Par, a parallel variant that autoregressively synthesizes all boundary frames $\mathcal{F}_i := \{F_i^{\text{begin}}, F_i^{\text{end}}\}, \forall i$ with frame self-improvement, then interpolates all video segments in parallel without video self-improvement. We compare A^{2}RD against SOTA autoregressive and parallel segment-based baselines: *(i) Direct Prompting*, which generates each segment independently from its description; *(ii) Naive Autoregressive (Naive-AR)*, which extrapolates each segment from the previous segment's last frame; *(iii) Naive-Par*, which autoregressively synthesizes segment end frames ($F_i^{\text{end}} := \text{TI2I}(S_i \mid F_1^{\text{begin}}, F_1^{\text{end}}, \ldots, F_{i-1}^{\text{end}})$) and interpolates each segment between its boundary frames; *(iv) MovieAgent* (Wu et al., 2025c), a hierarchical multi-agent framework for long-form movie generation; *(v) ViMax* (HKUDS, 2025), an end-to-end multi-agent framework for script planning, storyboarding, character design, and reference-guided shot synthesis; and *(vi) VideoMemory* (Zhou et al., 2026), which generates segments autoregressively using a dynamic entity memory bank for visual consistency. We also experiment with open-weight TI2V models and fully open-weight stacks in Section D.1.

Implementation Details. We run A^{2}RD with two iterations for frame and two for video refinements, resulting in at most 6 videos and 6 frames. For efficiency, we pre-compute Equations (2) and (3) and Equation (4) ($\text{MLLM}_{\text{rel}}^{\text{ref}}$) over all segments at once. MVMEM storage grows linearly with #segments and is limited according to available hardware. However, the context used by each operation remains bounded through relevance-based retrieval, and we cap the maximum input window at 100 for operations involving $\mathcal{S}$ and image lists (e.g., Equations (1) to (3)). See Section F for more details.

5.2 AUTOMATIC EVALUATIONS

Automatic Metrics. We evaluate generated videos across seven metrics: *(i) Semantic Alignment* measuring ViCLIP-based (Wang et al., 2024b) text-video similarity between each video segment and its description; *(ii) Narrative Coherence* following Wang et al. (2025a), where we employ Gemini 3.1 Pro (Google DeepMind, 2026) to score story, entity and environment progression on a scale of 0–1 with Self-Consistency (Wang et al., 2023), with penalties for repetitive or incoherent content. For inter-shot consistency, we measure *(iii) Character Consistency* and *(iv) Environment Consistency* across shots containing the same characters or environments, following An et al. (2026); Meng et al. (2026), and *(v) Motion Consistency*, customized from VBench, across the boundaries of temporally contiguous segments. These inter-shot metrics assess consistency across relevant shots, which may occur within or across generated segments. For intra-shot consistency, we report *(vi) Subject Consistency* and *(vii) Environment Consistency* from VBench. We assess general video quality in Section D.5, and provide more details in Section F and qualitative analysis in Section E.

Single-Scene Results. Table 1 presents our results on 1m generation on VBench-Long. The baselines reveal a tension between narrative progression and visual consistency, with Naive-AR achieving the highest baseline narrative coherence (0.75), while VideoMemory leads in consistency (0.57 character and 0.73 environment). For narrative coherence, we find that existing segment-based methods remain less competitive (0.69, ViMax and 0.67, VideoMemory), as they force shot changes at every segment which unfortunately produces repetitive environments in single-scene scenarios (Figure 5, Section I). Meanwhile, A^{2}RD achieves much better continuity thanks to its Adaptive Segment Generation and HITS, reaching 0.9 which is 20% better than the best baseline. For consistency, A^{2}RD attains significant gains in both characters (0.74 vs. 0.57 of VideoMemory, a 30% gain) and environments.

Method	Type	3-min (24 scenes)					5-min (40 scenes)				
		Semantic Alignment	Narrative Coherence	Inter-shot Character	Inter-shot Background	Inter-shot Motions	Semantic Alignment	Narrative Coherence	Inter-shot Character	Inter-shot Environment	Inter-shot Motions
Direct Prompting	Par	0.2116	0.3525	0.3123	0.2819	0.9856	0.1700	0.3026	0.3102	0.3083	0.9760
Naive-AR	AR	0.2044	0.6325	0.3519	0.3224	0.9791	0.1409	0.6088	0.3310	0.3950	0.9823
Naive-Par	Par	0.1941	0.8175	0.3637	0.3298	0.9813	0.1501	0.7132	0.3785	0.3869	0.9760
MovieAgent	Par	0.1886	0.5733	0.2922	0.2713	0.9513	0.1643	0.6112	0.3010	0.2569	0.9771
ViMax	Par	0.2108	0.8600	0.3557	0.3231	0.9756	0.1559	0.7750	0.3570	0.3838	0.9799
VideoMemory	AR	0.2062	0.8475	0.3687	0.3554	0.9759	0.1545	0.8292	0.3697	0.3830	0.9778
A^{2}RD-Par (Ours)	Par	0.2153	0.8641	0.4082	**0.4270**	0.9918	**0.1976**	0.8813	0.3930	**0.4567**	0.9926
A^{2}RD (Ours)	AR	**0.2215**	**0.9000**	**0.4262**	0.4148	**0.9941**	0.1806	**0.9500**	**0.4395**	0.4119	**0.9927**

Table 2: Experiments on LVBENCH-C (Ours) for multi-scene 3-min and 5-min video synthesis.

Notably, it obtains a motion consistency of 0.9935, substantially surpassing baselines and indicating that transitions between consecutive segments are nearly as smooth as a single diffusion generation. In addition, A^{2}RD-Par also performs competitively with 0.81 narratives while enabling parallel generation, offering a practical efficiency-consistency trade-off. Both A^{2}RD versions achieve the highest semantic alignment, suggesting that they better satisfy user prompts than baselines.

Multi-Scene Results. Table 2 reveals the critical challenge of long-horizon consistency, with all baselines degrading notably compared to Table 1. In both 3m and 5m settings, we see that baseline character consistency reaches only up to 0.38, while environment peaks at 0.39. Baselines also show significantly lower semantic alignment at 5m, highlighting the difficulty of maintaining storyline over extended horizons. Meanwhile, A^{2}RD achieves superior consistency, outperforming VideoMemory by 16% on average at 3m and 13% at 5m. It also produces notably more coherent narratives, scoring 10% higher than VideoMemory with a motion smoothness of above 0.99. Additionally, we also see that baseline narrative scores are higher on LVBENCH-C than VBench-Long. This is because LVBENCH-C focuses on multi-scene scenarios which better suit baselines that enforce regular shot transitions. Finally, our analysis over windows of up to 40 scenes shows that environment consistency is most challenging, with baselines generating similar but not identical spatial details (e.g., Section E.7). Character consistency lies in between, while object consistency is the least challenging one, which we hypothesize is easier to maintain through text descriptions (Figure 7).

Scaling Baselines. For fair comparisons, in Figure 3, we experiment with *best-of-N* (see Section F.5.1) scaling autoregressive baselines matching the same number of videos and frames sampled by A^{2}RD on VBench-Long, where we use Gemini 3.1 Pro for final selection. We find that consistency indeed improves remarkably for all baselines: Naive-AR improves from 0.61 $\rightarrow$ 0.67 avg. consistency, while VideoMemory improves from 0.65 $\rightarrow$ 0.71. However, narrative coherence is not always the case, with Naive-AR decreasing. Meanwhile, A^{2}RD shows significantly more promising test-time scaling potential thanks to its multi-dimensional judges, which can more reliably distinguish quality among candidates, improving from 0.73 $\rightarrow$ 0.78 for avg. consistency and 0.74 $\rightarrow$ 0.9 for coherence. Finally, our additional budget-matched experiments with refinement policies further show that neither best-of-N selection nor segment-only refinement matches A^{2}RD's gains (Section D.2).

5.3 HUMAN EVALUATIONS

Human Metrics. We conduct human studies on two datasets: 40 single-scene samples from VBench-Long and the three-minute multi-scene subset of LVBENCH-C, following a similar scale to Yu et al. (2025b). Seven highly qualified evaluators rate videos from all methods in randomized, anonymized order. Evaluators rate each video on six criteria on a 5-point Likert scale: *(i) Character Consistency*, *(ii) Object Consistency*, *(iii) Environment Consistency* (consistent layout across segments), *(iv) Transition Smoothness* (visual and temporal naturalness of cuts), *(v) Narrative Coherence* (logical story progression with causal relationships), and *(vi) Reference Consistency* (faithfulness to the global references). Each sample is rated by at least 2 evaluators and scores are averaged.

Human Results. Table 3 presents our human evaluation results on VBench-Long, while the corresponding results on LVBENCH-C are provided in Section D.6. As they exhibit the same trends, we discuss VBench-Long's results here. We see that these results corroborate our automatic metrics: A^{2}RD achieves the highest scores across all six criteria (avg. 4.68), substantially outperforming the best baseline, VideoMemory (3.93). It excels in character consistency (4.89) and narrative coherence (4.75), confirming strong identity preservation and story progression. It also scores 4.34 in transition smoothness, significantly higher than the best baseline of 3.63. Reference consistency reaches 4.91, showing potentially strong alignment when users provide reference images. While A^{2}RD-Par also maintains strong character consistency, it shows notable drops in transition smoothness and environ-

Method	Character Consistency	Object Consistency	Environment Consistency	Transition Smoothness	Narrative Coherence	Reference Consistency	Average
Naive-AR	3.95	4.05	3.85	3.63	4.11	N/A	3.92
Naive-Par	4.19	3.72	3.66	3.34	3.84	N/A	3.75
MovieAgent (Wu et al., 2025c)	3.40	2.80	2.77	2.20	2.39	3.20	2.79
ViMax (HKUDS, 2025)	4.23	3.95	3.41	2.64	3.59	4.50	3.72
VideoMemory (Zhou et al., 2026)	4.40	4.05	3.95	3.10	3.65	4.40	3.93
A^{2}RD-Par (Ours)	4.74	4.26	4.17	3.71	4.17	4.71	4.29
A^{2}RD (Ours)	**4.89**	**4.67**	**4.52**	**4.34**	**4.75**	**4.91**	**4.68**

Table 3: Human evaluation results on 40 VBench-Long samples (1–5 scale).

Method	Semantic Alignment	Narrative Coherence	Inter-shot Character	Inter-shot Environment	Inter-shot Motions
A^{2}RD-Par (Ours)	0.2094	0.8082	0.6817	0.8180	0.9843
A^{2}RD (Ours)	**0.2111**	**0.8987**	0.7353	0.8368	**0.9935**
A^{2}RD w/o MVMEM	0.1912	0.7235	0.5061	0.7392	0.9890
A^{2}RD w/o MVMEM's Text States	0.2032	0.7794	0.6791	0.7645	0.9903
A^{2}RD w/o MVMEM's Videos	0.2064	0.8647	0.7034	0.8153	0.9881
A^{2}RD w/o HITS	0.2054	0.7395	0.6774	0.7871	0.9911
A^{2}RD w/o MAPO	0.2095	0.7900	0.6843	0.8248	0.9904
A^{2}RD w/o Global References	0.2080	0.8136	0.7107	0.7923	0.9913
A^{2}RD always extrapolates	0.2054	0.8278	0.7112	0.8136	0.9912
A^{2}RD always interpolates	0.2084	0.7094	**0.7400**	**0.8499**	0.9924

Figure 2: Ablation studies over A^{2}RD's MVMEM, TTS algorithms, and adaptive segment generation strategies on Vbench-Long.

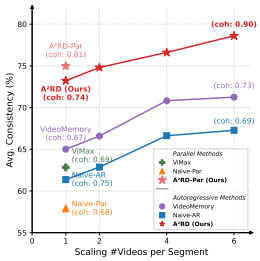


Figure 3: Consistency versus scaling #videos per segment.

ment consistency where errors arising in the middle of segments remain uncorrected. This confirms the benefits of autoregressive paradigm for both visual and temporal coherence.

5.4 Ablation Studies

Ablation Setups. We conduct ablation studies on VBench-Long to assess each A^{2}RD component's contribution across three groups: (i) MVMEM's components including the complete MVMEM system, its Textual States, and its Videos; (ii) Test-time scaling components including global references, HITS, and MAPO (replaced by Self-Refine (Madaan et al., 2023)); and (iii) adaptive segment generation strategy when we instead always extrapolate, or interpolate. We omit components that have been comprehensively studied in prior work, such as MVMEM's Frames.

Ablation Results. Figure 2 shows the critical role of each component. First, MVMEM is the backbone of A^{2}RD; removing it severely degrades performance to near Naive-AR levels, as the system loses long-range dependency conditioning, consistency validation, and HITS. Ablating MVMEM's individual modalities reveals their contributions: without Textual States, narrative and consistency drop notably, while removing Videos causes less impact, as they are primarily for motion continuity. Second, test-time scaling components are also critical: we find that removing HITS causes considerable drops ($0.90 \rightarrow 0.74$ for narratives, $0.74 \rightarrow 0.68$ for characters), while without MAPO, the prompt refinements are less effective. Interestingly, we see that removing global references minimally affects narrative and character consistency but notably drops environment consistency ($0.84 \rightarrow 0.79$), suggesting that environments are harder to maintain and benefit from global references. Finally, the adaptive mechanism is also crucial: always extrapolating maintains reasonable narrative coherence (0.83) but reduces consistency. Meanwhile, always interpolating achieves the highest consistency but at the cost of reduced narrative coherence. In this case, each frame is conditioned on richer context from previous segments, and HITS (Video) further enforce intra-shot consistency, which together can over-constrain the generation and tend to produce limited visual progression.

6 Conclusions

We presented A^{2}RD, an agentic autoregressive architecture for long video synthesis. By decoupling creative synthesis from consistency, A^{2}RD addresses two fundamental challenges: temporal consistency and narrative coherence. This is achieved through three key components: a multimodal video memory for cross-modal context tracking; an adaptive generation mechanism enabling natural narrative progression with consistency enforcement; and hierarchical test-time self-improving algorithms that self-refine frames and segments to prevent error propagation. We further introduced LVBENCH-C, a benchmark designed to stress-test long-horizon consistency via cyclical appearance with optional state evolutions. Experiments across 1-10 minute videos show that A^{2}RD sets a new state-of-the-art for narrative coherence and visual consistency compared to existing baselines.

REFERENCES

Zhaochong An, Menglin Jia, Haonan Qiu, Zijian Zhou, Xiaoke Huang, Zhiheng Liu, Weiming Ren, Kumara Kahatapitiya, Ding Liu, Sen He, et al. Onestory: Coherent multi-shot video generation with adaptive memory. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2026.

Anthropic. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5, 2025.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

ByteDance Seed. Seedance 2.0. https://seedance2.ai/, 2026.

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. In *The Fourteenth International Conference on Learning Representations*, 2026.

Brandon Castellano. PySceneDetect. URL https://github.com/Breakthrough/PySceneDetect.

Shuo Chen, Cong Wei, Sun Sun, Ping Nie, Kai Zhou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. *arXiv preprint arXiv:2602.06028*, 2026.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Mohamed Elmoghany, Ryan Rossi, Seunghyun Yoon, Subhojyoti Mukherjee, Eslam Mohamed Bakr, Puneet Mathur, Gang Wu, Viet Dac Lai, Nedim Lipka, Ruiyi Zhang, et al. A survey on long-video storytelling generation: architectures, consistency, and cinematic quality. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7023–7035, 2025.

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7346–7356, 2023.

Bingjie Gao, Xinyu Gao, Xiaoxue Wu, Yujie Zhou, Yu Qiao, Li Niu, Xinyuan Chen, and Yaohui Wang. The devil is in the prompts: Retrieval-augmented prompt optimization for text-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3173–3183, 2025.

Google Deepmind. Veo. https://deepmind.google/models/veo/, 2025.

Google DeepMind. Gemini 3 Flash: Best for frontier intelligence at speed. https://deepmind.google/models/gemini/flash/, 2025. Google DeepMind.

Google DeepMind. Gemini 3.1 pro. https://deepmind.google/models/gemini/pro/, 2026.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Haoran He, Jiajun Liang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, and Ling Pan. Scaling image and video generation via test-time evolutionary search. *arXiv preprint arXiv:2505.17618*, 2025.

HKUDS. ViMax: Agentic video generation. https://github.com/HKUDS/ViMax, 2025. GitHub repository.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

Susung Hong, Brian Curless, Ira Kemelmacher-Shlizerman, and Steve Seitz. Comic: Agentic sketch comedy generation. *arXiv preprint arXiv:2603.11048*, 2026.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, et al. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Yidong Huang, Zun Wang, Han Lin, Dong-Ki Kim, Shayegan Omidshafiei, Jaehong Yoon, Yue Zhang, and Mohit Bansal. Planning with sketch-guided verification for physics-aware video generation. *arXiv preprint arXiv:2511.17450*, 2025b.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025c.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3668–3678, 2015.

Fangfu Liu, Hanyang Wang, Yimo Cai, Kaiyan Zhang, Xiaohang Zhan, and Yueqi Duan. Video-t1: Test-time scaling for video generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 18671–18681, 2025.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *The Fourteenth International Conference on Learning Representations*, 2026.

Do Xuan Long, Xingchen Wan, Hootan Nakhost, Chen-Yu Lee, Tomas Pfister, and Sercan Ö Arık. Vista: A test-time self-improving video generation agent. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2026.

Yue Ma, Kunyu Feng, Zhongyuan Hu, Xinyu Wang, Yucheng Wang, Mingzhe Zheng, Bingyuan Wang, Qinghe Wang, Xuanhua He, Hongfa Wang, et al. Controllable video generation: A survey. *arXiv preprint arXiv:2507.16869*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

Oscar Mañas, Pietro Astolfi, Melissa Hall, Candace Ross, Jack Urbanek, Adina Williams, Aishwarya Agrawal, Adriana Romero-Soriano, and Michal Drozdzal. Improving text-to-image consistency via automatic prompt optimization. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. Featured Certification.

Yihao Meng, Hao Ouyang, Yue Yu, Qiuyu Wang, Wen Wang, Ka Leong Cheng, Hanlin Wang, Yixuan Li, Cheng Chen, Yanhong Zeng, et al. Holocine: Holistic generation of cinematic multi-shot long video narratives. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2026.

Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. Memgpt: towards llms as operating systems. 2023.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International conference on machine learning*, pp. 4055–4064. PMLR, 2018.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Xiangyan Qu, Zhenlong Yuan, Jing Tang, Rui Chen, Datao Tang, Meng Yu, Lei Sun, Yancheng Bai, Xiangxiang Chu, Gaopeng Gou, et al. From scale to speed: Adaptive test-time scaling for image editing. *arXiv preprint arXiv:2603.00141*, 2026.

Naina Raisinghani. Nano Banana 2: Combining Pro capabilities with lightning-fast speed. https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/, 2026.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2023.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yiwen Song, Tomas Pfister, and Yale Song. Vqqa: An agentic approach for video evaluation and quality improvement. *arXiv preprint arXiv:2603.12310*, 2026.

Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2025.

Qwen Team. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

Rejin Varghese and M Sambath. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International conference on advances in data engineering and intelligent computing systems (ADICS)*, pp. 1–6. IEEE, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025a.

Xingchen Wan, Han Zhou, Ruoxi Sun, Hootan Nakhost, Ke Jiang, Rajarishi Sinha, and Sercan Ö Arık. Maestro: Self-improving text-to-image generation via agent orchestration. *arXiv preprint arXiv:2509.10704*, 2025b.

Qian Wang, Ziqi Huang, Ruoxi Jia, Paul Debevec, and Ning Yu. MAViS: A multi-agent framework for long-sequence video storytelling. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2273–2295. Association for Computational Linguistics, 2026.

Qinghe Wang, Xiaoyu Shi, Baolu Li, Weikang Bian, Quande Liu, Huchuan Lu, Xintao Wang, Pengfei Wan, Kun Gai, and Xu Jia. Multishotmaster: A controllable multi-shot video generation framework. *arXiv preprint arXiv:2512.03041*, 2025a.

Ruochen Wang, Ting Liu, Cho-Jui Hsieh, and Boqing Gong. On discrete prompt optimization for diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Yu Qiao. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *The Twelfth International Conference on Learning Representations*, 2024b.

Yu Wang, Dmitry Krotov, Yuanzhe Hu, Yifan Gao, Wangchunshu Zhou, Julian McAuley, Dan Gutfreund, Rogerio Feris, and Zexue He. M+: Extending memoryLLM with scalable long-term memory. In *Forty-second International Conference on Machine Learning*, 2025b.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Weijia Wu, Mingyu Liu, Zeyu Zhu, Xi Xia, Haoen Feng, Wen Wang, Kevin Qinghong Lin, Chunhua Shen, and Mike Zheng Shou. Moviebench: A hierarchical movie level dataset for long video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28984–28994, 2025b.

Weijia Wu, Zeyu Zhu, and Mike Zheng Shou. Automated movie generation via multi-agent cot planning. *arXiv preprint arXiv:2503.07314*, 2025c.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for LLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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

Yu Yang, Yue Liao, Jianbiao Mei, Baisen Wang, Xuemeng Yang, Licheng Wen, Jiangning Zhang, Xiangtai Li, Hanlin Chen, Botian Shi, et al. Spiral: A closed-loop framework for self-improving action world models via reflective planning agents. *arXiv preprint arXiv:2603.08403*, 2026b.

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. *arXiv preprint arXiv:2512.05081*, 2025.

Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. Nuwa-xl: Diffusion over diffusion for extremely long video generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1309–1320, 2023.

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. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025a.

Yifei Yu, Xiaoshan Wu, Xinting Hu, Tao Hu, Yangtian Sun, Xiaoyang Lyu, Bo Wang, Lin Ma, Yuewen Ma, Zhongrui Wang, et al. Videossm: Autoregressive long video generation with hybrid state-space memory. *arXiv preprint arXiv:2512.04519*, 2025b.

Kaiwen Zhang, Liming Jiang, Angtian Wang, Jacob Zhiyuan Fang, Tiancheng Zhi, Qing Yan, Hao Kang, Xin Lu, and Xingang Pan. Storymem: Multi-shot long video storytelling with memory. *arXiv preprint arXiv:2512.19539*, 2025a.

Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Ziyu Guo, Haoquan Zhang, Manyuan Zhang, Jiaming Liu, Peng Gao, and Hongsheng Li. Let's verify and reinforce image generation step by step. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28662–28672, 2025b.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47, 2025c.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Min Zhao, Hongzhou Zhu, Yingze Wang, Bokai Yan, Jintao Zhang, Guande He, Ling Yang, Chongxuan Li, and Jun Zhu. Ultravico: Breaking extrapolation limits in video diffusion transformers. In *The Fourteenth International Conference on Learning Representations*, 2026.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Mingzhe Zheng, Yongqi Xu, Haojian Huang, Xuran Ma, Yexin Liu, Wenjie Shu, Yatian Pang, Feilong Tang, Qifeng Chen, Harry Yang, et al. Videogen-of-thought: Step-by-step generating multi-shot video with minimal manual intervention. *arXiv preprint arXiv:2412.02259*, 2024.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

Jinsong Zhou, Yihua Du, Xinli Xu, Luozhou Wang, Zijie Zhuang, Yehang Zhang, Shuaibo Li, Xiaojun Hu, Bolan Su, and Ying-cong Chen. Videomemory: Toward consistent video generation via memory integration. *arXiv preprint arXiv:2601.03655*, 2026.

Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5802–5810, 2019.

Tianrui Zhu, Shiyi Zhang, Jiawei Shao, and Yansong Tang. Kv-edit: Training-free image editing for precise background preservation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16607–16617, 2025.

Zihao Zhu, Ruotong Wang, Siwei Lyu, Min Zhang, and Baoyuan Wu. Brandfusion: A multi-agent framework for seamless brand integration in text-to-video generation. *arXiv preprint arXiv:2603.02816*, 2026.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15329–15339, 2025.

A^{2}RD: Agentic Autoregressive Diffusion for Long Video Consistency

(Appendix)

A Terminologies and Summary of Major Implemented Functions **16**

B Teaser Examples **16**

C Additional Discussions **16**
C.1 Limitations . . . 16
C.2 Customization and Personalization . . . 18
C.3 Ethic Statements, Reproducibility Statements, and Use of AI Assistants . . . 19
C.4 Extended Related Work . . . 19
C.5 Benchmark Comparisons . . . 20

D Additional Experiments **20**
D.1 Generalization Across Open-Weight Models . . . 20
D.2 Additional Experiments on Budget-Matched Refinement Policies . . . 20
D.3 Consistency Analysis over Extended Horizons . . . 21
D.4 Scaling to Longer-Horizon Video Generation . . . 22
D.5 General Video Quality Assessments . . . 22
D.6 Additional Human Evaluation Results . . . 22

E Methodology Design Analysis: Why Does A^{2}RD Work? **23**
E.1 Adaptive Segment Generation Mechanism . . . 23
E.2 Why Co-Optimization in Extrapolation Mode Is Crucial (Section 3.3) . . . 24
E.3 How A^{2}RD Maintains Long-Distance Motion Continuity ($\text{MLLM}^{\text{img}}_{\text{retr}}$ in Equation (5)) 24
E.4 The Role of MVMEM's Textual States . . . 25
E.5 Why Do We Need Those MLLM-Judge Criteria? . . . 25
E.6 Memory-Augmented Prompt Optimization (MAPO) . . . 26
E.7 Error Analysis . . . 27
E.8 Test-Time Self-Improvement Analysis . . . 27
E.9 Cost and Latency Analysis . . . 28

F Implementation Details **30**
F.1 A^{2}RD-Parallel (A^{2}RD-Par) Details . . . 30
F.2 Experimental Details . . . 30
F.3 Additional Dataset Construction Details . . . 30
F.4 Automatic Metrics' Implementation Details . . . 31
F.5 Human Metrics' Scoring Details . . . 32

G LVBENCH-C: Examples **33**

H Methodology Prompts **38**
H.1 MVMEM Initialization Prompts . . . 38
H.2 A^{2}RD Prompts . . . 39
H.3 HITS Self-Refinement Prompts: MLLM-Judge . . . 44
H.4 MAPO Prompts . . . 50
H.5 Section D.3's Prompts . . . 54

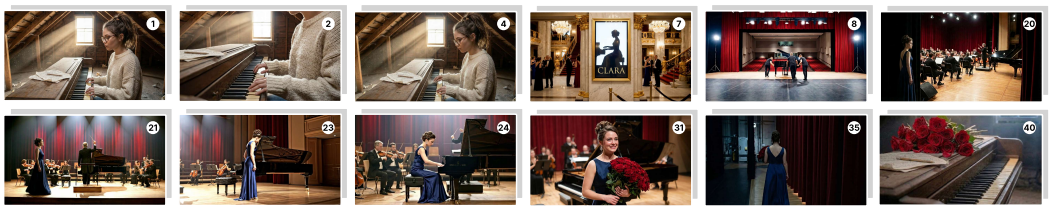
I Full Storyboard Examples **56**
I.1 A^{2}RD Examples . . . 56
I.2 Baseline Examples . . . 56

A Terminologies and Summary of Major Implemented Functions

Terminology	Description
Shot	A continuous sequence of frames captured from a single camera angle without cuts.
Scene	A narrative unit representing continuous action within a single physical environment or location.
Segment (Clip)	The fundamental generation unit in A^2RD, which is flexible and can span one or multiple shots or scenes.
Segment Context (S_i)	The textual description dictating the narrative, actions, and settings for the i-th segment.
Storyline ($\mathcal{S}$)	The complete sequential collection of segment contexts $\{S_1, \ldots, S_N\}$ defining the full video narrative.
Extrapolation	A generation mode that synthesizes a video segment moving forward from only a beginning frame.
Interpolation	A generation mode that synthesizes a video segment to seamlessly connect a fixed beginning and ending frame.

Table 4: Glossary of key terms used throughout the paper.

B Teaser Examples



Multi-Scene (LVBench-C; 5m): *A disheveled young pianist named Clara transforms from quietly composing in a dusty attic to delivering a triumphant, emotionally charged concert performance at a grand theater, ending the night before returning to her room and gently placing the flowers on the piano in gratitude.*



Single-Scene (VBench-Long; 1m): *Animated scene features a close-up of a short fluffy monster kneeling beside a melting red candle. The art style is 3D and realistic, with a focus on lighting and texture. The mood of the painting is one of wonder and curiosity, as the monster gazes at the flame...*



Single-Scene (VBench-Long; 1m): *A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually...*

Figure 4: Examples of 1m and 5m videos generated by A^{2}RD with Veo 3.1, showcasing consistent and coherent stories in static, dynamic, and multi-shot environments. Full and more storyboards are provided in Figures 5 and 6 and Section I, with videos in the supplementary materials.

Figures 4 to 6 present our teaser examples discussed in the main paper.

C Additional Discussions

C.1 Limitations

We acknowledge that A^{2}RD incurs more computational overhead than the baselines in Section 5. However, since all baselines follow the passive generation paradigm, direct comparison is therefore the meaningful way to evaluate their capabilities. To ensure fairness, we also provide compute-matched variants of all autoregressive baselines in the Scaling Baselines paragraph. In addition, it is worth noting that A^{2}RD's active reasoning incurs insignificant costs. Our analysis following

Function	Implemented Variants	Description	Reference	Prompt(s)
$\text{MLLM}_{\text{plan}}$	$\text{MLLM}^{\text{bg}}_{\text{plan}}$, $\text{MLLM}^{\text{ent}}_{\text{plan}}$	Plans prompts for establishing global reference backgrounds and entities.	Equation (1)	Section H.1
MLLM_{cap}	MLLM_{cap}	Generates captions for user-provided reference frames.	Equation (1)	Section H.1
MLLM_{dep}	MLLM_{dep}	Constructs a dependency graph to determine the synthesis order of global references.	Init (ii)	Section H.1
$\text{MLLM}_{\text{mode}}$	$\text{MLLM}_{\text{mode}}$	Adaptively determines the segment generation mode.	Equation (2)	Section H.2.1
$\text{MLLM}_{\text{retr}}$	$\text{MLLM}^{\text{scenes}}_{\text{retr}}$, $\text{MLLM}^{\text{cont}}_{\text{retr}}$, $\text{MLLM}^{\text{ref}}_{\text{retr}}$, $\text{MLLM}^{\text{img}}_{\text{retr}}$	Retrieves narratively relevant previous segments, contiguous contexts, and global references.	Equations (3) to (5)	Sections H.2.2 to H.2.5
$\text{MLLM}_{\text{pgen}}$	$\text{MLLM}^{\text{img}}_{\text{pgen}}$, $\text{MLLM}^{\text{vid}}_{\text{pgen}}$	Generates detailed narrative prompts for boundary frames and video segments.	Pipeline (iii) and (iv)	Sections H.2.6 and H.2.8
MLLM_{ext}	$\text{MLLM}^{\text{img}}_{\text{ext}}$, $\text{MLLM}^{\text{vid}}_{\text{ext}}$	Extracts fine-grained textual states from synthesized frames and videos.	HITS	Sections H.3.1 and H.3.2
$\text{MLLM}^{\text{img}}_{\text{vfy}}$	$\text{MLLM}^{\text{img-imgs}}_{\text{vfy}}$, $\text{MLLM}^{\text{img-st}}_{\text{vfy}}$, $\text{MLLM}^{\text{img-qual}}_{\text{vfy}}$	Verifies intra-consistency and basic quality for frames.	Equation (7)	Sections H.3.3 to H.3.6
$\text{MLLM}^{\text{vid}}_{\text{vfy}}$	$\text{MLLM}^{\text{vid-vid}}_{\text{vfy}}$, $\text{MLLM}^{\text{vid-qual}}_{\text{vfy}}$	Verifies inter- and intra-consistency and basic quality for videos.	Equation (9)	Sections H.3.7 and H.3.8
$\text{MLLM}^{\text{const}}_{\text{vfy}}$	$\text{MLLM}^{\text{const}}_{\text{vfy}}$	Evaluates consistency drift over extended horizons.	Equation (11)	Section H.5
MAPO	Feedback Reasoning, Lesson Synthesis, Prompt Refinement	Reason over judge outcomes, derive guidelines from past positive and negative refinement cases, and refine prompts targetedly	HITS	Section H.4

Table 5: Summary of main MLLM and synthesis functions implemented in A^2RD.

Section E.9 reveals that its cost overhead with using Gemini 3 Flash can be estimated only about less than 0.5$ per segment (with less than 10K tokens per call on average; excluding costs associated with generating 5 additional videos and 5 additional frames per segment).

In addition, we did not report human agreement scores in our study. This is because several videos are rated by exactly two reviewers, making the agreement not very informative. Additionally, evaluating long-form video generation is inherently subjective, particularly for complex criteria such as transition smoothness and narrative coherence. We decided to average scores over raters, as they are all highly qualified.

Finally, while A^2RD made solid progress in long-form video synthesis, several limitations remain. It requires component models (MLLM, TI2I, TI2V models) with strong instruction-following and visual reasoning capabilities. Additionally, the verification rubrics reflect implicit assumptions about consistency and quality that may not generalize to diverse creative styles, cultural contexts, or domain-specific preferences. We encourage adapting them for domain-specific settings. Finally, our user study reveals that transition smoothness and physical environment consistency remain the most challenging aspects for segment-based methods, presenting promising directions for future work.

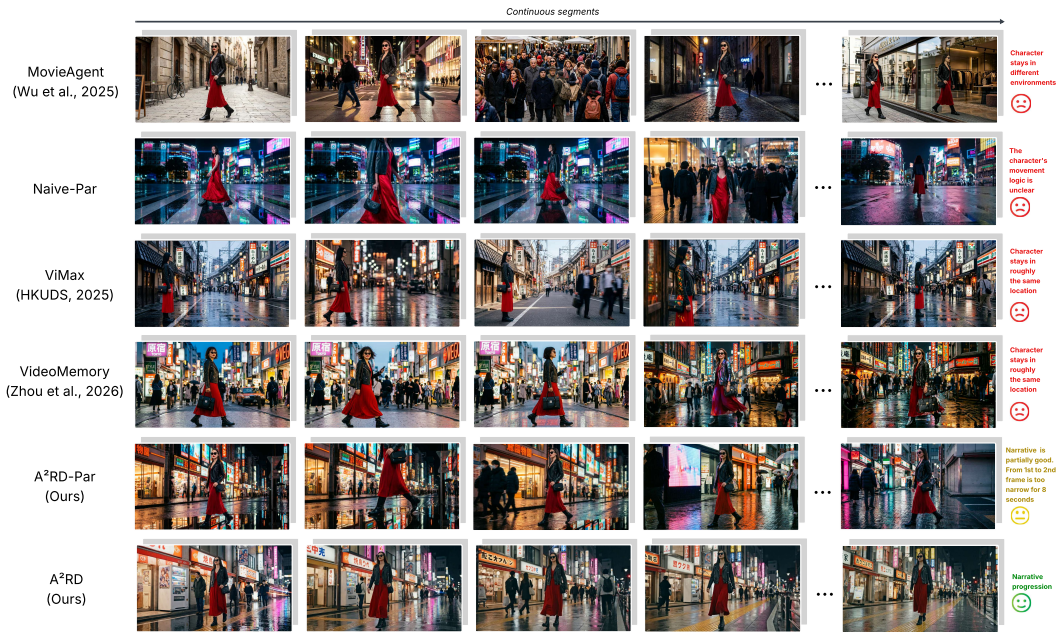


Figure 5: Narrative progression: a VBench-Long sample depicting a woman walking on a Japanese street. A^{2}RD maintains coherent and continue story, while baselines exhibit poor entity and environment progression.

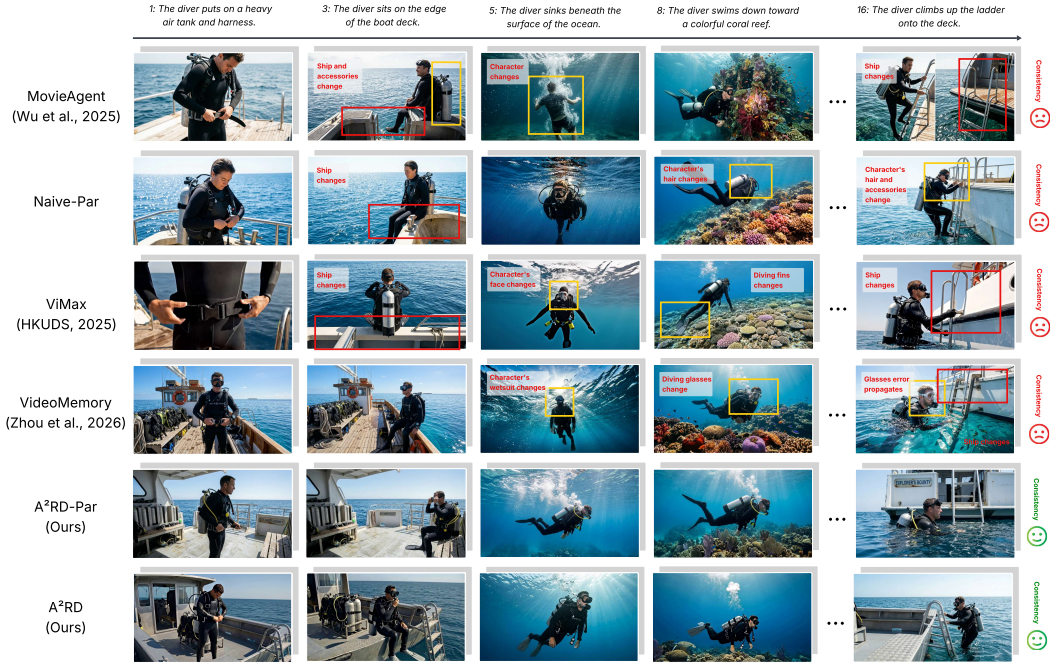


Figure 6: Long-horizon consistency: a LVBENCH-C sample depicting a diver preparing, diving, and returning to the ship. A^{2}RD maintains rigorous consistency while baselines suffer from unintended drifts in environments (red) and entities (yellow), such as changing character's hair, face, accessories and ship layouts.

C.2 CUSTOMIZATION AND PERSONALIZATION

A^{2}RD provides several practical forms of user control and customization. Specifically, users can edit the storyline before generation to specify the actions, settings, and cinematography of individual segments, and can provide reference images to control the appearance of characters, objects, and

Benchmark	Basic Statistics			Single-Scene			Multi-Scene			
	#Samples / #Prompts	Avg. Prompt Length	Avg. Dur.	Identity Consistency	Temporal Consistency	Spatial Consistency	Character State Evolving	Object State Evolving	Environment State Evolving	Cyclical Appearance
VBench (Huang et al., 2025c)	946 / 946	7.64	-	✓	✓	✓	✗	✗	✗	✗
VBench-Long (Huang et al., 2025c)	40 / 40	48.25	-	✓	✓	✓	✗	✗	✗	✗
T2V-CompBench (Sun et al., 2025)	1400 / 1400	10.42	-	✓	✓	✓	✗	✗	✗	✗
VBench-2.0 (Zheng et al., 2025)	1013 / 1013	21.83	-	✓	✓	✓	✗	✗	✗	✗
MovieGenBench (Polyak et al., 2024)	1003 / 1003	18.21	-	✓	✓	✓	✗	✗	✗	✗
MovieBench (test) (Wu et al., 2025b)	6 / 2875	91.5	32.0m	✓	✓	✓	-	-	-	-
MoviePrompts (Wu et al., 2025c)	10 / 10	95.70	-	✓	✓	✓	✓	✗	✗	✗
ST-Bench (Zhang et al., 2025a)	30 / 95	40.69	1m	✓	✓	✓	✓	✓	✓	✗
LVBENCH-C (Ours)	125 / 5200	18.25	3m, 5m, 10m	✓	✓	✓	✓	✓	✓	✓

Table 6: Comparison of text-to-video generation benchmarks. Existing benchmarks focus on single-scene consistency but lack evaluation of the challenging cyclical state tracking where entities and environments appear, disappear for extended periods, then reappear with non-trivial evolved states.

environments. The verification criteria are also customizable: beyond our default consistency and quality criteria, users may add requirements for visual style, brand adherence, personalized character or product attributes, or preferred cinematographic composition. We leave these directions for future work.

C.3 Ethic Statements, Reproducibility Statements, and Use of AI Assistants

Use of AI Assistants. The authors used AI-assisted tools (ChatGPT, Gemini, and Claude) for coding and writing support. All substantive content, analysis, and conclusions remain solely the authors' own work.

Ethics Statements. This work follows the ICLR Code of Ethics. Our human evaluation involves rating generated videos and does not collect sensitive personal information. LVBENCH-C contains synthetically constructed text scenarios and is manually reviewed to reduce harmful, biased, or inappropriate content. Nevertheless, improving long-form video synthesis may lower the barrier to creating convincing synthetic media, including deepfakes and misinformation. Generated content may also inherit social or cultural biases from the underlying models. We therefore encourage responsible use alongside appropriate content safeguards, provenance mechanisms, and synthetic-media detection.

Reproducibility Statements. We provide detailed descriptions of A^{2}RD, its implemented functions, model configurations, refinement settings, evaluation metrics, and latency analysis in the main paper and appendices. The appendices further include the complete prompts used for memory initialization, retrieval, generation, verification, and prompt optimization; construction details and examples for LVBENCH-C; human-evaluation guidelines; baseline configurations; and additional qualitative and quantitative results. Our sample generated videos and their full storyboards are included in the supplementary materials. We will release the code and LVBENCH-C if applicable to support future research.

C.4 Extended Related Work

Memory for LLM Agents. Memory has become an important component in modern agentic systems, enabling agents to maintain long-range dependencies across sequential decisions (Hu et al., 2025; Zhang et al., 2025c). Current memories for LLM agents save information in diverse formats including text (Packer et al., 2023; Zhong et al., 2024), hidden representations (Wang et al., 2025b), and graphs (Chhikara et al., 2025; Xu et al., 2025), and typically incorporate retrieval mechanisms (e.g., semantic search) alongside management strategies (e.g., updating). Memory construction for image and video synthesis has also been increasingly studied, where the memory is typically composed of images (Parmar et al., 2018; Zhang et al., 2025a; Yu et al., 2025a; Zhou et al., 2026), image–text pairs (Zhu et al., 2019), and hidden representations (Zhu et al., 2025; Cai et al., 2026). While image-based memories provide visual references, relying on the generative models to implicitly infer entity identity and narrative state is unreliable over long horizons. Representation-based memories offer seamless conditioning but lack interpretability for explicit consistency control. A^{2}RD addresses both limitations with a multimodal memory that explicitly tracks fine-grained visual and narrative progression across modalities, enabling targeted control over consistency and coherence.

Model / Method	Semantic Alignment	Narrative Coherence	Inter-shot Character	Inter-shot Environment	Inter-shot Motion
LTX-Video 0.9.8 (13B) (HaCohen et al., 2024)					
Naive-AR	0.1884	0.5903	0.5040	0.5788	0.9558
A^{2}RD (Ours)	0.1978	0.7922	0.7011	0.7719	0.9852
A^{2}RD, Fully Open-Weight (Ours)	0.1944	0.7002	0.6244	0.7263	0.9723
Wan 2.2 (5B) (Wan et al., 2025a)					
Naive-AR	0.1584	0.6719	0.5104	0.5898	0.9752
A^{2}RD (Ours)	0.2219	0.8000	0.6912	0.7824	0.9901
A^{2}RD, Fully Open-Weight (Ours)	0.1982	0.7250	0.5861	0.7393	0.9817

Table 7: Experimental results across TI2V backbones and model stacks. A^{2}RD-Open replaces the default MLLM and TI2I models with Qwen3.5-27B and Qwen-Image-2512, respectively.

C.5 Benchmark Comparisons

Table 6 compares our benchmark against representative existing studies. Since these studies do not explicitly report the attributes we evaluate, we downloaded their publicly available datasets and analyzed the samples ourselves to determine the presence or absence of each criterion.

D Additional Experiments

D.1 Generalization Across Open-Weight Models

We examine the generalization of A^{2}RD across open-weight models.

Setups. To examine whether the gains of A^{2}RD depend on a particular video generator or proprietary agent components, we evaluate it with two open-weight TI2V backbones: LTX-Video 0.9.8 (13B) (HaCohen et al., 2024) and Wan 2.2 (5B) (Wan et al., 2025a). For each backbone, we evaluate both the default A^{2}RD stack, which uses Gemini 3 Flash as the MLLM and Nano Banana 2 as the TI2I model, and a fully open-weight stack that replaces them with Qwen3.5-27B (Team, 2026) and Qwen-Image-2512 (Wu et al., 2025a), respectively. For both models, we use 30 denoising steps at 704×480 resolution and follow the same evaluation protocol on VBench-Long in Section 5. Since Wan 2.2 does not support interpolation, we run all A^{2}RD experiments with this backbone using the always-extrapolate mode.

Results. As shown in Table 7, A^{2}RD improves over Naive-AR across both models. On LTX-Video, it yields notable gains in narrative coherence ($0.59 \rightarrow 0.79$) and character consistency ($0.50 \rightarrow 0.70$). Wan 2.2 follows a similar trend, with narrative coherence rising from $0.67 \rightarrow 0.80$, character consistency from $0.51 \rightarrow 0.69$, and environment consistency from $0.59 \rightarrow 0.78$. With the fully open-weight LTX-Video stack, narrative coherence improves from 0.5903 to 0.7002, character consistency from 0.5040 to 0.6244, and environment consistency from 0.5788 to 0.7263. The fully open-weight Wan 2.2 stack exhibits the same trend, improving narrative coherence from 0.6719 to 0.7250 and environment consistency from 0.5898 to 0.7393. Although the fully open-weight variants perform below their default counterparts, they retain consistent gains over Naive-AR without changing the method or its hyperparameters. These results show that A^{2}RD benefits from stronger underlying components but is not tied to a particular TI2V backbone or proprietary model stack.

D.2 Additional Experiments on Budget-Matched Refinement Policies

We examine whether budget-matched candidate selection or segment-only refinement can achieve A^{2}RD's gains without cross-segment feedback, which therefore highlights the advantages of our HITS algorithms.

Setups. We compare three refinement policies using Qwen3.5-27B as the MLLM, Qwen-Image-2512 as the TI2I model, and either LTX-Video 0.9.8 or Wan 2.2 as the TI2V backbone. *Search-only* generates candidates without feedback and selects the best using A^{2}RD's judge in Equation (9). *Segment-only refinement* follows the same frame and video refinement schedule as HITS, but evaluates

Refinement Policy	Semantic Alignment	Narrative Coherence	Inter-shot Character	Inter-shot Environment	Inter-shot Motion
LTX-Video 0.9.8 (13B) (HaCohen et al., 2024)					
Naive-AR	0.1884	0.5903	0.5040	0.5788	0.9558
Search-only	0.1852	0.6508	0.5642	0.6236	0.9707
Segment-only refinement	0.1932	0.6075	0.6025	0.6591	0.9592
A^{2}RD, Fully Open-Weight (Ours)	0.1944	0.7002	0.6244	0.7263	0.9723
Wan 2.2 (5B) (Wan et al., 2025a)					
Naive-AR	0.1584	0.6719	0.5104	0.5898	0.9752
Search-only	0.1665	0.7050	0.5301	0.6095	0.9822
Segment-only refinement	0.1819	0.6693	0.5537	0.6607	0.9794
A^{2}RD, Fully Open-Weight (Ours)	0.1982	0.7250	0.5861	0.7393	0.9817

Table 8: Refinement-policy comparison using fully open-weight model stacks.

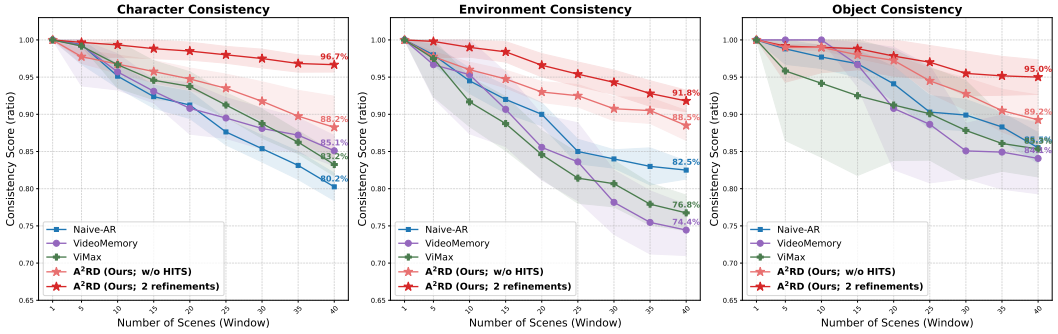


Figure 7: Consistency scores as a function of scene window size (1–40). All methods are evaluated on LVBENCH-C, 5-minute using LLM-Judge. A^{2}RD (Ours) consistently maintains higher consistency over extended horizons compared to baselines.

and revises each segment using only its prompt, boundary frames, and generated video, without retrieved states or content from earlier segments. A^{2}RD*, Fully Open-Weight (Ours)* is our own method. The three policies receive the same number of image and video generation calls per segment, with early stopping disabled. Wan 2.2 uses extrapolation for all policies because it does not support interpolation. Naive-AR is included as a reference baseline. The full-method values are from our earlier experiments in Table 7.

Results. Table 8 presents our results, which reveal interesting trade-offs across methods. Overall, we observe that Search-only improves narrative coherence and inter-shot motion over Naive-AR with both backbones, suggesting that A^{2}RD's cross-segment judge is indeed helpful in selecting candidates with better story progression and smoother transitions. However, this baseline offers limited ability to correct visual errors, and its semantic alignment even decreases slightly with LTX-Video. Meanwhile, Segment-only refinement achieves higher semantic alignment and inter-shot character and environment consistency than Search-only, showing the benefit of actively revising boundary frames and video segments. Yet its narrative coherence does not benefit similarly for both LTX-Video (0.6075 vs. 0.6508) and Wan 2.2 (0.6693 vs. 0.7050), suggesting that without retrieved past states and references during refinement, improving a segment locally does not necessarily preserve its place in the broader story. Overall, by refining with our cross-segment context, A^{2}RD achieves the highest alignment, narrative coherence, and character and environment consistency with both backbones.

D.3 Consistency Analysis over Extended Horizons

Setups. We analyze how consistency degrades as generation extends over longer horizons. Since the automatic metrics in Section 5.2 cannot fully capture the evolving states of entities and environments, we develop an MLLM-Judge method to evaluate consistency in these evolving scenarios. Specifically, after grouping segments that share the same characters, objects, or environments for automatic metrics as in Section F.4, a carefully calibrated MLLM-Judge with a state-of-the-art MLLM (Google DeepMind, 2026) (Section H.5 for prompts) evaluates each dimension $d \in \{\text{char}, \text{obj}, \text{env}\}$ per segment through 3 steps: *(i) Identify Visual Differences* by comparing with relevant segments' frames

Method	Subject Consistency	Background Consistency	Aesthetic Quality	Imaging Quality	Temporal Flickering	Motion Smoothness
Direct Prompting	0.7713	0.8929	0.6180	0.7160	0.9638	0.9915
Naive-AR	0.9062	0.9363	0.6546	0.7424	0.9800	0.9898
Naive-Par	0.8852	0.9148	**0.6692**	0.6417	0.9873	**0.9924**
MovieAgent	0.9150	0.9386	0.6532	**0.7450**	0.9641	0.9914
ViMax	0.9215	0.9488	0.6181	0.7040	0.9668	0.9914
VideoMemory	0.9207	0.9492	0.6435	0.7029	0.9907	0.9901
A^{2}RD-Par (Ours)	0.9108	**0.9578**	0.6209	0.7257	**0.9910**	0.9917
A^{2}RD (Ours)	**0.9231**	0.9514	0.6632	0.7431	0.9578	0.9901

Table 9: General video quality over six VBench metrics (Huang et al., 2024).

and references; *(ii) Classify Inconsistencies* as expected (justified by the story) or unexplained; and *(iii) Flag Inconsistencies* that are unexplained and obvious. For each sample i-th, with the same notation introduced in Section 3 and $\mathbb{1}$ being the indicator function, the d's consistency ratio is:

$$\text{Consistency}^d = \frac{\Sigma_{i=1}^{|\mathcal{S}|} \mathbb{1}\left[\text{MLLM}_{\text{vfy}}^{\text{const}}(S_i, F_i^{\text{begin}}, \mathcal{F}_{\text{ref}}, \mathcal{S}_{\text{ref}}, d)\right]}{|\mathcal{S}|} \tag{11}$$

Results. Figure 7 reveals interesting insights. For baselines, we see that they exhibit a monotonic decline in consistency as #scenes grows, confirming that consistency in long horizons is a non-trivial challenge. For characters, Naive-Par performs worse than VideoMemory and ViMax, which is expected given their dedicated character memory mechanisms. Environment consistency proves the most challenging dimension overall. Interestingly, Naive-Par performs much better than ViMax and VideoMemory here; we attribute this to these baselines forcing frequent shot changes, which causes more often environment hallucinations. Finally, all baselines perform similarly on object consistency, the least challenging dimension, likely because objects do not exhibit complex identities compared to characters and environments. Our method, A^{2}RD, outperforms baselines across all three axes by large margins: it retains 96.7% character consistency, 91.8% environment consistency, and 95.0% object consistency. These results further validate the superiority of A^{2}RD architecture in combating long-horizon consistency degradation.

We manually verified a subset of the MLLM-Judge outputs and found its flags to be effective (80% agreement) for detecting obvious errors. Figure 16 shows representatives. We want to note that subtle inconsistencies, which are very common in synthesized frames, may be missed—by design, as it is calibrated to flag only clear violations. Nevertheless, the method effectively captures the overall trend that A^{2}RD significantly outperforms baselines, aligning with other automatic and human evaluations.

D.4 Scaling to Longer-Horizon Video Generation

We experiment with A^{2}RD on ten 10-min scenarios from LVBench-C, approaching the frontier of current long-form video generation. By using the MLLM-Judge method developed above, on average, A^{2}RD attains average consistency of 90.5% for characters, 84.0% for environments, and 91.5% for objects, confirming its strong capability in generating coherent ultra-long videos.

D.5 General Video Quality Assessments

Table 9 presents the evaluation results on six VBench metrics that assess general video quality. Overall, A^{2}RD performs competitively across all metrics, achieving the best score in Subject Consistency and the second-best in Background Consistency, Aesthetic Quality, and Imaging Quality. The parallel variant, A^{2}RD-Par, obtains the highest scores in Background Consistency and Temporal Flickering. These results confirm that our method does not sacrifice visual fidelity for temporal consistency and narrative coherence.

D.6 Additional Human Evaluation Results

Following the protocol in Section 5.3, we further evaluate three-minute multi-scene videos from LVBench-C. As shown in Table 10, A^{2}RD achieves an average score of 4.39, outperforming

Method	Character Consistency	Object Consistency	Environment Consistency	Transition Smoothness	Narrative Coherence	Reference Consistency	Average
Naive-AR	3.35	3.19	3.32	3.61	3.77	N/A	3.45
Naive-Par	3.62	3.21	3.19	3.01	3.20	3.68	3.32
VideoMemory	3.80	3.15	3.38	3.16	3.80	4.02	3.55
A^{2}RD (Ours)	**4.77**	**4.22**	**3.92**	**4.31**	**4.63**	**4.68**	**4.39**

Table 10: Human evaluation results on the three-minute multi-scene subset of LVBENCH-C (1–5 scale). The average is computed over the available criteria for each method.

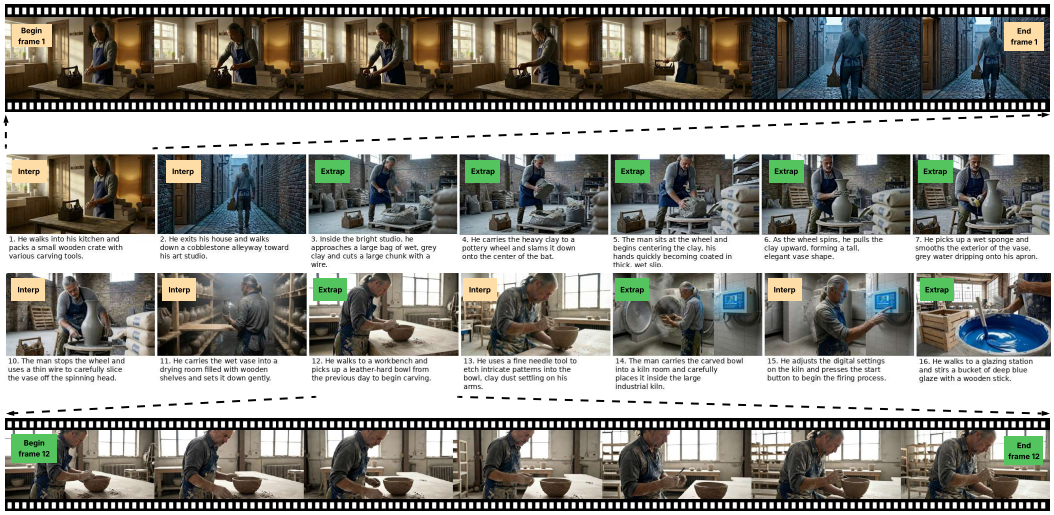


Figure 8: An example from LVBENCH-C about the story of a ceramist illustrating how our Adaptive Segment Generation (Equation (2)) determines the appropriate generation mode.

Naive-AR (3.45) and Naive-Par (3.32), while attaining 4.77 in character consistency and 4.63 in narrative coherence.

E METHODOLOGY DESIGN ANALYSIS: WHY DOES A^{2}RD WORK?

We justify why A^{2}RD works by analyzing its novel components, including Adaptive Segment Generation, Frame-Video Co-Optimization, MVMEM's Textual States, some criteria in MLLM-Judges, and MAPO. Components such as MVMEM's keyframes, and other MLLM-Judge criteria have been widely studied in existing works, so we omit them in this section. We denote red boxes as erroneous and green boxes as positive.

E.1 ADAPTIVE SEGMENT GENERATION MECHANISM

Adaptive Segment Generation is a core mechanism of A^{2}RD that balances natural narrative progression with consistency enforcement. Here we illustrate an example from LVBENCH-C where A^{2}RD determines the generation mode (Equation (2)) and why it is effective. In scene context 1, the character packs small items; since the next scene context 2 takes place in a Cobblestone Alleyway, which is a known, established environment during memory initialization, A^{2}RD selects Interpolation mode to enforce strict consistency with that setting. In contrast, in scene context 12, the character begins carving, and in scene 13 he continues the same action; since the next environment is unknown, A^{2}RD selects Extrapolation mode to allow the scene to evolve naturally from the current context, which preserves the natural progression of the room as shown in the keyframes from segment 12.

In practice, we find that state-of-the-art MLLM (Google DeepMind, 2025) performs this task reliably, achieving $> 85\%$ agreement with human experts; misdeterminations mostly involve determining extrapolation as interpolation. We note that such misdeterminations are largely benign, as the image model can sometimes successfully depict highly consistent environments in these cases.

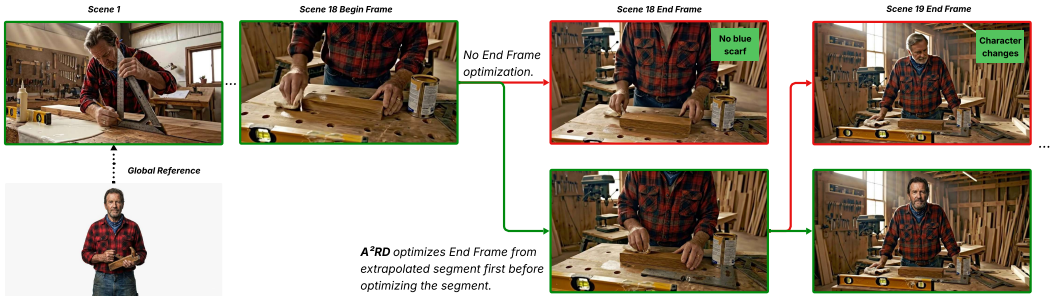


Figure 9: Qualitative comparison of with and without A^{2}RD's end frame co-optimization in extrapolation mode. Without co-optimization, correcting inconsistencies such as a missing blue scarf (Scene 18) and incorrect character identity (Scene 19) persist can be challenging. With A^{2}RD's co-optimization, these issues are corrected naturally.

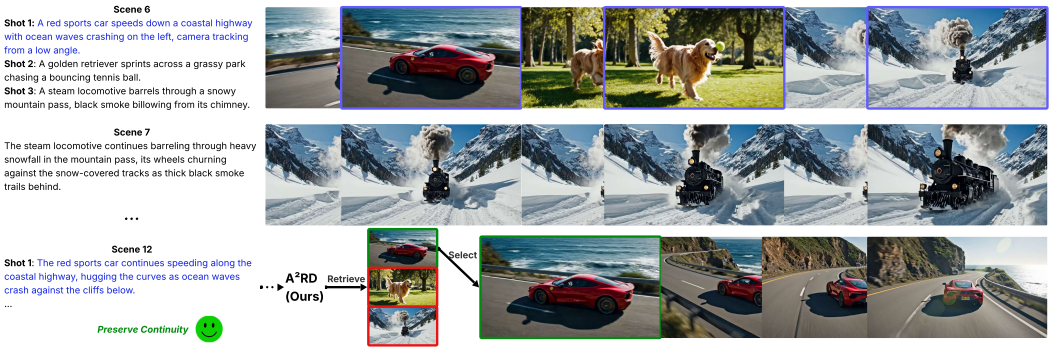


Figure 10: Qualitative comparison how A^{2}RD is able to maintain distanced continuity.

E.2 Why Co-Optimization in Extrapolation Mode Is Crucial (Section 3.3)

We present two examples illustrating why co-optimization in HITS is important when F_i^{end} is unavailable, as shown in Figure 9. The scenario depicts a carpenter working in a woodworking workshop as the camera gradually zooms in and out. We observe that without end frame optimization, the Scene 18 end frame is used directly and the prompt is only optimized to bridge the two frames which miss the blue scarf as an inconsistency. While this blue scarf can sometimes be sufficiently specified in the prompt to produce a similar one during video prompt optimization phase, the situation is more challenging for the Scene 19 end frame, where the character's face in the erroneous frame closely resembles but does not match the person in the reference frame. In that case, A^{2}RD demonstrates its superiority by correcting the character's face to maintain identity consistency.

E.3 How A^{2}RD Maintains Long-Distance Motion Continuity (MLLM$^{\text{IMG}}_{\text{RETR}}$ in Equation (5))

MLLM$^{\text{img}}_{\text{retr}}$ is one of our key contributions for bridging long-distance motion continuity, which, to the best of our knowledge, has not been addressed by prior works. Here, we explain how it works.

Consider the example in Figure 10 where Scene 6 contains three shots: a red sports car speeding down a coastal highway, a golden retriever sprinting across a grassy park, and a steam locomotive barreling through a snowy mountain pass. In Scene 7, we continue the locomotive's journey through the snow. Since A^{2}RD only saves the begin and end frames of each segment, the car's end-of-shot frame from Scene 6 is not directly available. Later, in Scene 12, we return to the red sports car, now continuing drifting around a sharp cliffside bend. To handle this, A^{2}RD extracts all end-of- shot frames from the stored video segment of Scene 6, then uses MLLM$^{\text{img}}_{\text{retr}}$ to retrieve the correct one (Shot 1 of Scene 6) and uses it as the begin frame of Scene 12, thereby preserving visual and motion continuity across the large temporal gap.

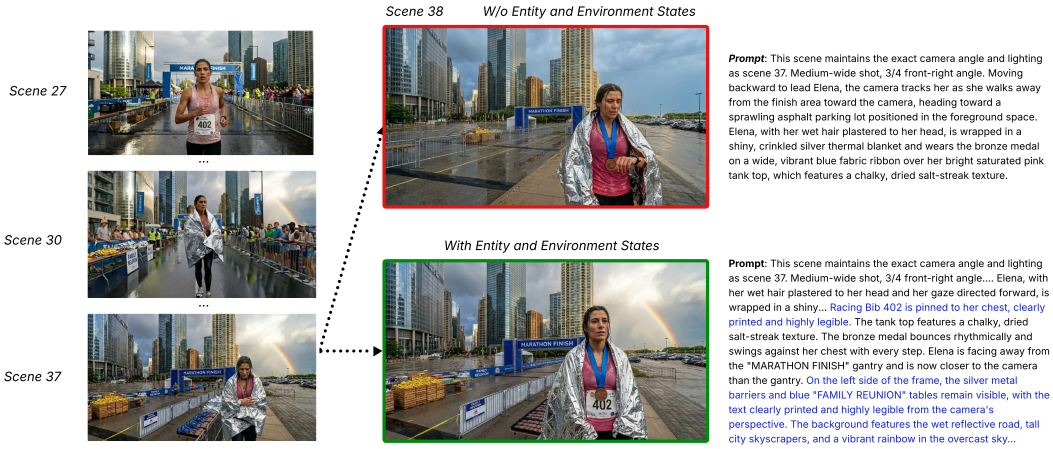


Figure 11: An example about the marathon of Elena showing the importance of MVMEM's Textual States. With Textual States, key details are explicitly enforced into the prompt, leading to highly consistent outcomes.

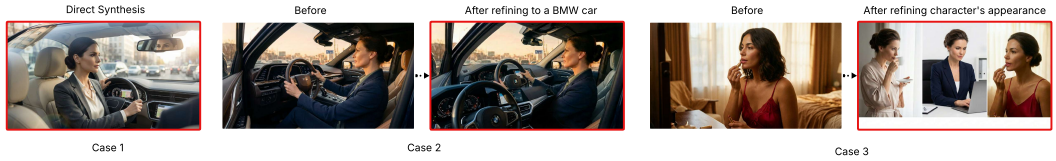


Figure 12: An example showing the importance of the Physical Plausibility criterion during frame synthesis and self-improvement. A synthesized frame with physical implausibilities can cause the resulting video to hallucinate at least at the beginning.



A²RD's *MLLM-Judge*: *"The image captures all elements of the prompt with high fidelity... The scene is very coherent and realistic, representing a single unified moment... Progression Discussion: The frame shows a clear and logical narrative progression from Scene 1 to Scene 2... Spatial Analysis: The current scene contains a significant spatial contradiction regarding the orientation of the vehicle when compared to the established environment in Scene 0. In Scene 0, the white car is parked in the driveway facing to the road, exposing its left (driver's) side to the camera. However, in the current scene, the vehicle has been flipped 180 degrees and is now facing to the left of the frame,* ***a change not justified by the scene descriptions****..."*

Figure 13: An example showing the importance of the Spatial Logicness criterion during frame self-improvement. Scene 3 passes all other criteria yet introduces a spatial discontinuity across segments.

E.4 THE ROLE OF MVMEM'S TEXTUAL STATES

Textual states are important for explicitly enforcing necessary details into frame and video prompts for consistency. Figure 11 shows an example for frame synthesis. We observe that while the prompt for Scene 38 without entity and environment states is quite comprehensive, several important details were not explicitly specified, leading to the synthesized frame having several inconsistencies. In particular, `Racing Bib 402` disappears after about 9 scenes, yet it needs to reappear when Elena's lower abdomen is visible. Other minor details such as the `FAMILY REUNION tables` and the `rainbow` are also lost. With MVMEM's Textual States, these components are explicitly retrieved and enforced into the prompt, together with reference frames from previous scenes and global anchors, leading to highly consistent outcome preserving these details.

E.5 WHY DO WE NEED THOSE MLLM-JUDGE CRITERIA?

From Section 3.3, we see that A²RD requires 8 metrics to verify frames and 10 metrics to verify videos. While some are intuitively important, such as Instruction Following and Inter/Intra Consis-

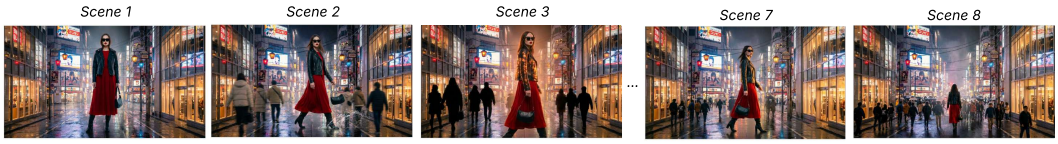


Figure 14: An example of a woman walking on a Japanese street, showing that without the Narrative Progression criterion during judging and self-improvement, the resulting storyboard can be highly consistent yet lack meaningful story progression.

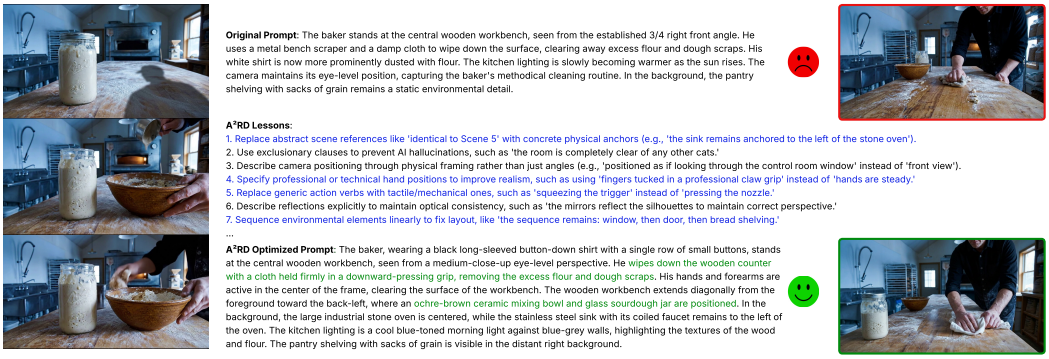


Figure 15: An example of MAPO in action. Given the original prompt and validation feedback, MAPO retrieves similar refinement cases from memory, synthesizes actionable lessons, and applies them to produce a refined prompt that directly addresses the failure modes, improving the average score from 6.4 to 8.3.

tency, the importance of others is less obvious, including Physical Plausibility, Spatial Logicalness, and Narrative Progression. Here, we dive into these metrics:

Physical Plausibility (Frame). Since we enforce consistency, synthesized and refined frames can become awkward, where the image model attempts to satisfy the consistency requirement without regard for image quality, as shown in Figure 12. Using these frames as the beginning frame for video synthesis leads to visual hallucinations, at least at the start of the video.

Spatial Logicalness (Frame). Spatial logicalness is an important criterion for maintaining continuity across segments. As shown in Figure 13, Scene 3 achieves perfect scores in all consistency and basic quality metrics. However, synthesizing a segment from Scene 2 and Scene 3 introduces a spatial discontinuity, as the car appears rotated 90 degrees. This breaks the natural continuity between the two scenes, potentially causing hallucination artifacts during video synthesis when interpolating between these frames.

Narrative Progression (Frame and Video). Balancing narrative progression and consistency requires the Narrative Progression criterion. Without it, we can end up with a storyboard that is highly consistent in both entity and environment, as shown in Figure 14, yet the story is not meaningful.

E.6 Memory-Augmented Prompt Optimization (MAPO)

Here, we present an example showing how MAPO works. The original prompt describes a baker who uses a metal bench scraper and a damp cloth to wipe down the surface seen from the established 3/4 right front angle. This prompt contains several points to be improved, such as the camera angle references an abstract established view and not a concrete physical description, and crucially background elements lack spatial ordering.

A^{2}RD's MAPO retrieves 10 positive and 5 negative cases and synthesizes 12 applicable lessons, notably: (1) *Replace abstract references with concrete physical anchors*, (4) *Specify professional hand positions*, (5) *Replace generic action verbs with tactile/mechanical ones*, and (7) *Sequence environmental elements linearly*. The refined prompt applies lesson (1) to anchor the foreground with

a glass sourdough jar on the front-left corner of the workbench and an ochre-brown ceramic mixing bowl in the back-left as concrete physical anchors, and to replace the established 3/4 right front angle with a medium-close-up eye- level perspective; applies lessons (4) and (5) to change the action to wipes down the wooden counter with a cloth held firmly in a downward-pressing grip with hands and forearms active in the center of the frame. The refined frame scores 8.3 on average, with entity reference consistency, environment reference consistency, and spatial logicalness all reaching 10/10, demonstrating that the retrieved lessons directly addressed the failure modes.

E.7 Error Analysis



Figure 16: Error analysis of A^{2}RD. Green boxes highlight temporally consistent frames, while red boxes indicate inconsistencies detected by our MLLM-Judge (Section D.3). We present representative failure cases spanning environment and entity consistency, suggesting that advancing foundation model reasoning and physics-aware control are promising directions for future work.

Environment Inconsistencies. Environment consistency remains the most challenging dimension, as layout and physical arrangements are challenging to control during image synthesis. We manually select two representative failure cases flagged by our MLLM-Judge (Section D.3), as shown in Figure 16: in Environment Case 1, the generator produces scenes visually similar but not physically identical to the reference; in Environment Case 2 (Scene 8), a wooden table abruptly appears when the server brings coffee to the customer. Both errors involve either (1) approximating rather than faithfully reproducing the physical layout, or (2) hallucinating physical elements. In addition, we observe that while the MLLM-Judge can effectively identify these issues, the test-time self-refinement loop with our current budget of two refinement iterations is not always sufficient to resolve them. Addressing these limitations via advancing foundation models with more precise physics-aware control will be a promising direction for future work.

Entity Inconsistencies. We provide a common entity error in Figure 16, where the image generator reproduces the same checkered shirt but with a slightly different color than the original reference. We found that the MLLM-Judge also occasionally fails to detect such minor inconsistencies, suggesting that similar errors could go unnoticed during self-refinement. Addressing this limitation will likely require both more accurate (MLLM-)judges and stronger reasoning foundation models.

E.8 Test-Time Self-Improvement Analysis

Iteration	Group 1: Consistency over Images				Group 2: Consistency over Texts			Group 3: Basic Quality		Avg.
	Entity Ref. Consistency	Env. Ref. Consistency	Narrative Progression	Spatial Logicalness	Character State	Object State	Environment State	Instruction Following	Physical Plausibility	
Init	9.250	9.000	9.125	7.083	9.667	8.375	8.708	8.875	8.583	8.741
1	9.667	9.958	9.708	8.375	9.717	9.083	9.208	8.833	8.583	9.259
2	**9.750**	**9.958**	**9.792**	**8.708**	**9.875**	**9.375**	**9.542**	**8.875**	**8.792**	**9.407**

Table 11: Frame-level MLLM-Judge scores across HITS refinement iterations averaged over all benchmarks.

Iteration	Group 1: Inter Consistency					Group 2: Intra Consistency			Group 3: Basic Quality			Avg.
	Inter Entity Consistency	Inter Environment Consistency	Inter Motion Consistency	Camera Consistency	Narrative Progression	Character State	Object State	Environment State	Instruction Following	Physical Plausibility	Frame Fit	
Init	7.250	7.430	7.425	7.225	8.700	9.875	9.725	9.450	8.325	8.175	9.625	7.928
1	7.525	8.112	7.775	7.475	9.000	9.850	9.825	9.600	8.575	8.600	9.800	8.194
2	**7.725**	**8.795**	**8.050**	**7.950**	**9.450**	**9.800**	**9.950**	**9.775**	**8.950**	**9.000**	**9.975**	**8.493**

Table 12: Video-level MLLM-Judge scores across HITS refinement iterations averaged over all benchmarks.

Setups. We examine the effectiveness of our HITS algorithms in refining frames and videos. We record the MLLM-Judge scores during experiments across two refinement iterations on all benchmarks. These scoring criteria are carefully calibrated by our experts during development.

Results. At the frame level, we find that most dimensions start off strong, with spatial logicalness (7.083) and physical plausibility (8.583) as the weakest dimensions. After two HITS iterations, both improve noticeably with spatial logicalness jumps from 7.083 $\rightarrow$ 8.708, while physical plausibility also sees a steady gain (8.583 $\rightarrow$ 8.792). Other dimensions such as entity reference consistency and narrative progression also benefit, pushing the overall average from 8.741 $\rightarrow$ 9.407. At the video level, inter-consistency metrics start relatively lower but improve steadily across iterations, with environment consistency showing the largest gain (7.430 $\rightarrow$ 8.795). Narrative progression and physical plausibility also benefit noticeably (8.700 $\rightarrow$ 9.450 and 8.175 $\rightarrow$ 9.000). Intra-consistency dimensions such as character, object, and environment state already start strong and converge near perfect scores by iteration 2. Overall, the average video score improves from 7.928 $\rightarrow$ 8.493, showing that HITS indeed enhances both temporal coherence and consistency. While MLLM scoring could be noisy, the relative trends align with both human judgments (Table 3) and automatic metrics (Table 6): motion and environmental consistency emerge as the weakest dimensions across all evaluations, and removing HITS notably drops coherence and consistency scores.

E.9 Cost and Latency Analysis

Let L_M, L_I, and L_V denote the latency of a single MLLM call, TI2I call, and TI2V call, respectively. Let N_{interp} ($N_{\text{interp}} \leq N$) denote the #interpolation segments. Our estimation analysis is under ideal hardwares, where the batch inference latency is expectedly equal to the one call. Naive-AR, as defined in Section 5, incurs approximately NL_M+NL_V, while Naive-Par incurs NL_M+$(N$+$1)L_I$+L_V.

E.9.1 Baseline Latency

MovieAgent (Wu et al., 2025c). MovieAgent is a passive parallel segment-based method that generates a movie by decomposing a script into scenes and shots, then synthesizing images and videos per shot. It decomposes a script hierarchically across three LLM-driven planning stages: Script breakdown, scene planning, and shot creation, followed by per-shot image and video synthesis. Each stage issues one LLM call per unit (sub-script, scene, and shot, respectively). Let N_s, N_c, and N denote the number of sub-scripts, scenes, and shots, and assuming all frames and videos can be synthesized in parallel (even though in the official implementation the authors process them sequentially), the total ideal latency is estimated as:

$$L_{\text{MovieAgent}} \approx (4+N_s+N_c)L_M+L_I+L_V \tag{12}$$

VideoMemory (Zhou et al., 2026). VideoMemory is an autoregressive segment-based method with a dynamic memory bank. It first calls StoryboardAgent (once) with 1 MLLM call to decompose synopsis into N shots. For each shot i sequentially:

- MemoryAgentAnalyze (per shot): 1 MLLM call (L_M) to decide REUSE or CREATE for each entity. For each entity that needs CREATE: 1 TI2I call (L_I). Let E_i = number of new entities in shot i, so this costs L_M+E_iL_I per shot.
- VisualizeAgentKeyframe (per shot): 1 MLLM call (L_M) to build prompts, then 1 TI2I call (L_I) for the keyframe, then 1 TI2V call (L_V) for the video. Cost: L_M+L_I+L_V per shot.

The sum of E_i terms is bounded by the total number of unique entities across all shots. Assume it is $|\mathcal{R}|$ for compatibility. The total ideal latency is estimated as,

$$L_{\text{VideoMemory}} \approx (1{+}2N)L_M{+}(N{+}|\mathcal{R}|)L_I{+}NL_V \tag{13}$$

E.9.2 A^{2}RD LATENCY

We omit the case $t \neq i-1$ (Equation (5)) in this analysis for simplicity.

MVMEM Init. This step incurs $2L_M$ from 2 batch MLLM calls: one for batch inference obtaining $\mathcal{P}_{\mathcal{R}}^{\text{bg}}$, $\mathcal{P}_{\mathcal{R}}^{\text{ent}}$, $\mathcal{P}_{\mathcal{R}}^{\text{u}}$ in Equation (1) and one for identifying dependencies (step (ii)). It incurs at most $|\mathcal{R}| \times L_I$ for synthesizing global references, thus the maximum init latency is $L_{\text{init}} \approx 2L_M{+}|\mathcal{R}|L_I$.

Precomputations. As mentioned in Section 5.1, we precompute all retrieval steps in Equations (3) and (4), and the segment generation mode in Equation (2), which incur 1 batch MLLM call L_M.

Synthesize and Self-Improve–Boundary Frame(s). A^{2}RD incurs an initial latency of $(N_{\text{interp}}{+}1)L_M$ for generating frame prompts ($P_0^{\text{begin}}/P_i^{\text{end}}$ in Equation (5)). Each prompt is subjected to a maximum of k_f refinement iterations. During each, A^{2}RD first synthesizes a batch of candidate frames via a TI2I call (L_I), extracts the textual states T_i^F via a batched MLLM call (L_M), and performs verification using another batched MLLM call (L_M). The best candidate then undergoes at most three subsequent MLLM calls ($3L_M$) for test-time optimization (allocated for feedback reasoning, lesson generation, and prompt refinement, respectively). In the final iteration, the best candidate frame is returned without optimization. Consequently, the approximated maximum refinement latency per segment is $k_f(L_I{+}2L_M){+}(k_f-1)3L_M$, which simplifies to $k_f(5L_M{+}L_I)-3L_M$. Scaling this across all interpolated segments and including the initial prompt generation, the total latency for boundary frame processing is approximated at most as $L_{\text{frame}} \approx (N_{\text{interp}}{+}1)\,[k_f(5L_M{+}L_I)-2L_M]$.

Synthesize and Self-Improve–Video Segment. Similarly, we can derive the latency of video segment synthesis. A^{2}RD incurs an initial latency of NL_M for generating the video prompts (P_i in Equation (6)). Each prompt is subjected to a maximum of k_v refinement iterations. During each, A^{2}RD first synthesizes a batch of candidate video segments via a TI2V call (L_V), extracts the video states T_i via a batched MLLM call (L_M), and performs verification using another batched MLLM call (L_M). The best candidate is then selected to undergo a refinement process contingent upon the generation mode. For interpolation, the agent utilizes one subsequent MLLM calls ($3L_M$) for test-time prompt optimization (allocated for feedback reasoning, lesson generation, and prompt refinement, respectively). For extrapolation, A^{2}RD incurs an additional computational cost to extract and *edit* the end frame. Because editing solely invokes feedback reasoning without lesson generation and prompt refinement, this frame edit iteration requires only ($3L_M{+}L_I$). Skipping the reasoning step in the final edit iteration, the extrapolation edit overhead becomes $k_f(3L_M{+}L_I)-L_M$. In the final iteration, the best candidate video is returned directly without the optimization process. Aggregating these across all segments and incorporating the extrapolation overhead, the latency is approximated as $L_{\text{video}} \approx N\,[k_v(5L_M{+}L_V)-2L_M]{+}(N-N_{\text{interp}})\,[k_f(3L_M{+}L_I)-L_M]$.

A^{2}RD Latency. In total, $L_{\text{A}^2\text{RD}} = L_{\text{init}}{+}L_{\text{pre}}{+}L_{\text{frame}}{+}L_{\text{video}}$, which can be simplified as:

$$L_{\text{A}^2\text{RD}} \approx [N(5k_v{+}3k_f-3){+}N_{\text{interp}}(2k_f-1){+}5k_f{+}1]\,L_M{+}(|\mathcal{R}|{+}Nk_f{+}k_f)L_I{+}(Nk_v)L_V \tag{14}$$

A^{2}RD-Par Latency. In A^{2}RD-Par, we set $N_{\text{interp}} = N$ and $k_v = 1$. Because video synthesis is fully parallelized, its latency does not scale with N and instead incurs $L_M{+}L_V$. Thus,

$$L_{\text{A}^2\text{RD-Par}} \approx [N(5k_f-2){+}5k_f{+}2]\,L_M{+}(|\mathcal{R}|{+}Nk_f{+}k_f)L_I{+}L_V \tag{15}$$

Compared to Equation (13), A^{2}RD incurs at most $\frac{5k_v+5k_f-6}{2}$ additional MLLM calls, k_f-1 additional TI2I calls, and k_v-1 additional TI2V calls per segment due to iterative self-improvement. In practice, we set $k_v = k_f = 3$ (one initial generation followed by two refinement iterations), resulting in roughly 12 additional MLLM calls, 2 additional TI2I calls, and 2 additional TI2V calls per segment compared to VideoMemory. With observed latencies of under 10 seconds per MLLM call, 30 seconds per TI2I call (Nano Banana), and under 120 seconds per TI2V call (Veo 3.1), A^{2}RD incurs a total computational overhead of under 7.2 minutes per segment.

Summary of Latency. As discussed above, A^{2}RD incurs a total computational overhead of under 7.2 minutes per segment. Nevertheless, this added latency is highly tractable when long-video consistency and coherence are the priority. As discussed in Section 5, none of the baselines maintains strong consistency over longer horizons; also see Figures 27 and 28 for examples. Even with a matched video-generation budget, best-of-N scaling does not close the gap. Achieving comparable quality with A^{2}RD therefore requires human manual trial and error, including inspecting segments, revising prompts, and regenerating frames and clips. A^{2}RD thus offers a cost-efficient and automatic quality-oriented solution for settings where very high multi-minute consistency and coherence matter more than latency. Nevertheless, reducing the latency of verification and refinement remains important future work, and A^{2}RD can benefit much more as foundation models get stronger.

Summary of Reasoning Cost. Our internal record reveals that the MLLM reasoning powering A^{2}RD is less than $0.50 per segment with Gemini 3 Flash. In addition, A^{2}RD generates up to six frame and six video candidates per segment, which incurs these image and video generation costs.

F Implementation Details

F.1 A^{2}RD-Parallel (A^{2}RD-Par) Details

We introduce A^{2}RD-Par, a parallel version of A^{2}RD, for efficiency. A^{2}RD-Par synthesizes both boundary frames $\mathcal{F}_i := \{F_i^{\text{begin}}, F_i^{\text{end}}\}, \forall i$ autoregressively and performs the same frame self-improvement process. All video segments are then synthesized in parallel, with no video self-improvement applied. In A^{2}RD, video synthesis latency takes Nk_vL_V, where N is the #segments, k_v is the #self-improvement iterations, and L_V is the latency of a single video synthesis call. A^{2}RD-Par removes the sequential dependency, reducing this to L_V under ideal hardware (Section E.9). While this sacrifices some spatial consistency for evolving environments, A^{2}RD-Par still enforces strict character consistency and produces coherent stories for scenes with static environments (Figures 5 and 6 for examples).

F.2 Experimental Details

We run A^{2}RD with two iterations for frame and two for video refinements. At each iteration, we synthesize two frames and two videos via batch inference, run all the judges in Equations (7) and (9) in parallel and the best one will be selected for next refinement iteration. We implement early stopping when average frame/video scores $\geq$ 9/10. In total, a video segment requires at most 6 videos and 6 images. For efficiency, we pre-compute Equations (2) and (3) and Equation (4) (MLLM$_{\text{rel-imgs}}$) over all segments at once, see *All Scenes* prompts in Section H.2. For scaling to longer horizons, we limit each MVMem schema to available hardwares, and cap the maximum window size at 100 for operations involving $\mathcal{S}$ and image lists as MLLM inputs (e.g., Equations (1) to (3)).

F.3 Additional Dataset Construction Details

Here, we provide more details of how we construct LVBench-C.

Human-In-The-Loop Dataset Construction. We instantiate LVBench-C with 3-, 5-, and 10-minute scenarios using a common time-independent construction pipeline. For each challenge type, we first carefully craft a professional screenwriter persona prompt, incorporating one human-designed demonstration and generate scenarios using a state-of-the-art MLLM (Google DeepMind, 2025). This generation is enforced with constraints: *(i) Content*, where story flow must be meaningful and natural, each segment fits a pre-defined duration, clear cause-and-effect relationships, no random events; *(ii) Gap Rules*, where main entities and environments must be absent for at least $n = 10$ segments before reappearing with or without state changes; and *(iii) State Change*, ensuring natural state changes through realistic activities, specific visual markers (positions, appearances, conditions). During generation, we manually review generated scenarios and update constraints or demonstrations to improve diversity.

Data Refinement and Deduplication. Since raw MLLM-generated scenarios often contain duplicates, logical gaps, and low-quality content, we apply systematic refinement. The generated scenarios are deduplicated using the same MLLM: we summarize these scenarios one by one, and

feed summaries into the MLLM to identify and remove similar scenarios. We then customize Self-Refine (Madaan et al., 2023) for self-refinement: selected scenarios undergo rigorous MLLM-Judge validation against six criteria: *(i) Specificity Verification* ensuring each scene must be specific enough with clear revelations; *(ii) Logic Verification* where each scene reveals new details not mentioned before that must have logically existed from beginning; *(iii) Natural Verification* where entity actions must be natural without forced or contrived scenarios; *(iv) Realism Verification* ensuring details must be realistic and appropriate with everyday believable activities; *(v) Repetition Verification* ensuring activities and details must be varied without repetitive content; and *(vi) Contradiction Verification* with no contradictions across segments regarding entity and environment states. Failed scenarios in any criterion undergo refinement until successful or up to a limited number of iterations. To avoid self-preference bias, a separate MLLM (Anthropic, 2025) re-verifies all scenarios using the same criteria, and refines any minor issues. We manually review a subset of samples to confirm quality.

F.4 Automatic Metrics' Implementation Details

- *Semantic Alignment* is computed using ViCLIP (Wang et al., 2024b), measuring cosine similarity between video and scene description embeddings.
- *Narrative Coherence* is evaluated by an MLLM prompted with the original story and the generated video. See the prompt below.
- *Inter-Shot Character and Environment*: we first group N scenes into two types of groups via the MLLM: groups having the same characters with the same states, groups having the same environments with the same states. Once these groups are established, we evaluate the *Inter-Shot Character* and *Inter-Shot Environment* consistency. We use YOLOv8 (Varghese & Sambath, 2024) to segment character and background regions within the $n = 3$ keyframes of each segment within each group, extracting their corresponding DINOv3 (Siméoni et al., 2025) embeddings. We then compute the pairwise cosine similarities of these embeddings across scenes within each group, and average the results across all groups to obtain the final metrics.
- *Inter-Shot Motions*: For each scene, we first identify its previous continuous scene (if exists), see Section H.2.2. For each couple, we then evaluate the transitions between the two synthesized video segments by extracting the last 10 frames of the preceding video segment and the first 10 frames of the subsequent video segment. We then compute the VBench `motion_smoothness` score across these boundary frames.
- *Intra-shot Consistency*: We use those metrics from VBench.

F.4.1 Prompt for Narrative Coherence Evaluation

```
You are evaluating the narrative coherence of a video story.

Context (For Reference Only): {story}

Watch the video and evaluate its narrative coherence on a scale of 0 to 1, where:
- 0.9-1.0 = Almost perfect: consistent story, character, object, and environment progression
- 0.6-0.8 = Good: story events follow a logical progression with clear cause-and-effect, but
minor inconsistencies in character appearance, object continuity, or environment identity are
present
- 0.3-0.5 = Moderate: story flow is partially coherent but noticeable inconsistencies in
character appearance, object continuity, or environment break immersion, or scenes feel
loosely connected without clear cause-and-effect
- 0.0-0.2 = Poor: story flow is largely incoherent with major inconsistencies across
character, object, or environment, or the narrative is broken, incomprehensible, or highly
repetitive without story justification

Consider and PENALIZE heavily for:
1. Story progression - What's wrong with the story flow from scene to scene? Do events follow
logically from prior events? Are cause-and-effect relationships between scenes clear and
believable? Penalize if scenes feel disconnected or outcomes appear without plausible causes.
2. Character progression - What's wrong with character appearance or identity progression?
Does the character's state or condition change causally as a result of story events?
3. Object progression - What's wrong with object progression across scenes? Do objects
appear, change, or disappear in ways that are causally justified by the story?
4. Environment progression - What's wrong with the setting progression? Are environment
changes causally motivated by the story rather than arbitrary?
5. Repetitive penalties - If repetitive activities or environments appear in the video but
are NOT present in the Context, the score MUST NOT exceed 0.6. If the Context itself
specifies repetitive actions or settings, do not penalize for repetition.
```

```
IMPORTANT: Heavily penalize any character appearance progression, object progression issues,
or environment shifts that break immersion. If the video, character, or environment as a
whole makes no sense, is imperceptible, or is incomprehensible, the score MUST NOT exceed
0.2.

Output ONLY a JSON object wrapped in ```json and ```:
```json
{
  "narrative_coherence": <score>,
  "reasoning": "details of the reasoning"
}
```
```

F.4.2 PROMPT FOR GROUPING SCENES FOR INTER-SHOT METRICS

```
Analyze this video scenario and its scenes to identify which scenes share the same
background/location and which scenes need the same character or object appearance (visual
consistency).

Scenario: {scenario}

Scenes:
{chr(10).join(f"{i}. {scene}" for i, scene in enumerate(scenes))}

Output JSON with:
- "background_groups": array of arrays, each inner array contains scene indices (0-based)
that share the same background/location
- "character_groups": array of arrays, each inner array contains scene indices (0-based)
where the same character should have consistent appearance (same face, body type, cloths,
core features)
- "object_groups": array of arrays, each inner array contains scene indices (0-based) where
the same object should have consistent appearance (same color, shape, material)

Example: {{"background_groups": [[0,1,2], [3,4]], "character_groups": [[0,1,3], [2,4,5]],
"object_groups": [[0,2,4]]}}

Output only the JSON object:
```

F.5 HUMAN METRICS' SCORING DETAILS

Thank you for participating in our human evaluation study! This human evaluation will take approximately 30 minutes. You will be presented with a randomly sampled subset of generated videos from all methods in **randomized, anonymized order**. Each video consists of multiple **8-second segments**. For each video, please watch it, review the reference images and scene descriptions, then submit your ratings. Please rate each video on **six criteria** using a **5-point Likert scale**:

Score	Meaning
1	Very poor
2	Poor
3	Acceptable/Ok to Watch
4	Good
5	Excellent

EVALUATION CRITERIA

1. **Character Consistency**: Do characters maintain consistent appearance across video?
2. **Object Consistency**: Do objects maintain consistent appearance across video?
3. **Environment Consistency**: Do backgrounds and environments remain consistent across transitions?
4. **Transition Smoothness**: Are the cuts between segments visually and temporally natural?
5. **Narrative Coherence**: Does the story progress logically with meaningful causal relationships?
6. **Reference Consistency**: How faithfully does the generated video adhere to the provided reference images? N/A if no reference images are provided.

F.5.1 BEST-OF-N BASELINE'S PROMPTS

```
You are an expert image quality evaluator. Given {n} candidate keyframe images generated from
the same scene description and reference entities, select the single best candidate.

Scene Description: {scene_description}

Reference entities:
[reference_image_parts]

Candidates:
[candidate_image_parts]

Evaluate each candidate on the following criteria:
1. Scene faithfulness { how well the image matches the described scene
2. Visual quality { sharpness, composition, absence of artifacts
3. Entity consistency { how closely characters and objects match the reference images

Select the candidate that best satisfies all criteria holistically.
Respond ONLY with JSON: {"best": <1-{n}>, "reason": "<brief justification>"}
```

```
You are an expert video quality evaluator. Given {n} candidate video clips generated from the
same prompt and previous clip, select the single best candidate.

Scene Description: {scene_description}

Previous clip:
[previous_video_part]

Candidates:
[candidate_video_parts]

Evaluate each candidate on the following criteria:
1. Prompt faithfulness { how well the video matches the described scene
2. Visual quality { sharpness, color accuracy, absence of artifacts
3. Motion naturalness { smooth, physically plausible continuation from previous clip

Select the candidate that best satisfies all criteria holistically.
Respond ONLY with JSON: {"best": <1-{n}>, "reason": "<brief justification>"}
```

G LVBENCH-C: EXAMPLES

Scenes 1-5 (Dutch oven & Chef appear - Initial cooking):

1. A heavy cast-iron Dutch oven sits empty and cold on a gas stove.
2. A chef pours golden olive oil into the Dutch oven as the flame ignites below.
3. Chopped onions and garlic are tossed into the Dutch oven, sizzling in the hot oil.
4. Slabs of raw beef are added to the Dutch oven, browning quickly against the metal.
5. A splash of red wine is poured into the Dutch oven, deglazing the bottom as steam rises.

Scenes 6-15 (Chef transitions to prep - Dutch oven absent):

6. The chef walks to the pantry to grab a bag of fresh organic carrots.
7. He peels the carrots over a compost bin with quick, rhythmic strokes.
8. The carrots are sliced into thick medallions on a heavy wooden cutting board.
9. A bundle of fresh thyme and rosemary is tied together with kitchen twine.
10. The chef cleans his professional knife carefully under a stream of warm water.
11. He sets the dining table with linen napkins and polished silver cutlery.
12. Two crystal wine glasses are placed precisely next to the dinner plates.
13. A crusty baguette is sliced and placed into a decorative wicker bread basket.
14. The chef checks his watch, noting the time remaining for the slow-cooking process.
15. He wipes down the marble countertop until it shines under the bright kitchen lights.

Scenes 16-20 (Return to Dutch oven - Serving):

16. The Dutch oven is now filled with a thick, bubbling beef stew and tender vegetables.
17. The chef lifts the lid of the Dutch oven, releasing a dense cloud of savory steam.
18. He ladles the rich stew from the Dutch oven into a large ceramic serving bowl.
19. The Dutch oven is moved to a heat-proof mat, its exterior now stained with dried drips.
20. He sprinkles fresh parsley over the stew inside the Dutch oven before serving.

Scenes 21-24 (Dining room - Final scene):

21. Guests enter the dining room, reacting to the rich aroma of the cooked meal.
22. The chef carries the serving bowl to the table as guests take their seats.
23. Everyone begins to eat, enjoying the deep flavors developed over several hours.
24. The chef smiles as he watches his friends enjoy the hearty homemade dinner.

Figure 17: **Example 3 minute (24 scenes) scenario from LVBENCH-C, Object State Evolving**: The Dutch oven appears in (1-5), disappears in (6-15), then reappears in (16-20).

Scenes 1-4 (Elias & Sing appear - Initial state):

1. Elias and Sing lounge on a stained sofa wearing torn undershirts and mismatched flip-flops.
2. Sing slams the table, shouting that they are destined for greatness, not noodles.
3. Elias looks down at his empty bowl, a spark of sudden, desperate greed in his eyes.
4. Elias grabs Sing's collar and yells that they must find the Magic Master to change their lives.

Scenes 5-14 (Characters absent - 10 scenes):

5. A wide shot reveals a room thick with expensive cigar smoke where gamblers shout and shove chips.
6. The Rich Street Boy walks in, slamming a stack of heavy gold bars onto the green felt table.
7. The boy screams a challenge at the empty dealer's chair, his voice echoing through the hall.
8. The camera pans to the top of the grand stairs, revealing the Master with a cigarette dangling from his lip.
9. The Master descends the staircase slowly, the smoke trailing behind him like a silk ribbon.
10. He stops halfway, leaning over the gold-leaf railing to stare down at the Street Boy.
11. The Master reaches the table and sits, the leather chair creaking under his weight of authority.
12. The Master spreads a card deck in a perfect, lightning-fast rainbow arc across the felt.
13. The Rich Street Boy bluffs, sweat dripping off his chin as the Master stares him down.
14. With a flick of his wrist, the Master reveals the winning card, ending the game instantly.

Scenes 15-40 (Characters reappear - State changed):

15. Elias and Sing stand by a pillar in the room, now wearing oversized, poorly-fitted tuxedos with crooked ties.
16. Sing tries to look dignified but accidentally trips over his own overly-long trouser hem.
17. Elias whispers urgently, his face pale and eyes twitching with desperate hope.
18. The duo walks toward the Master's table, bowing so low their foreheads nearly hit the floor.
19. The Master looks at the duo and flicks his cigarette ash directly onto Elias's shoe.
20. Sing opens his mouth to speak but the Master raises one finger, silence falls instantly.
21. The Master deals three cards face-down, then looks at them with complete disinterest.
22. Elias reaches for a card but the Master slaps his hand away without even looking.
23. The Rich Street Boy snickers and tosses a single coin at Sing's feet mockingly.
24. Sing's face flushes red with shame, his fists clenching at his sides.
25. The coin rolls across the floor, everyone's eyes following it in tense silence.
26. It stops at the Master's foot. He crushes it flat.
27. Sing suddenly drops to both knees, forehead touching the floor in a full kowtow.
28. Elias hesitates, then joins him, both men prostrated before the Master's chair.
29. The entire casino goes silent, even the roulette wheel stops spinning.
30. The Master stands up slowly, his chair scraping loudly against the marble floor.
31. He walks around them in a circle, examining them like livestock at a market.
32. The Master stops, picks up the flattened coin from under his shoe.
33. He flips it high into the air without warning.
34. Sing's eyes track the coin. He lunges and catches it mid-air with desperate speed.
35. The Master's expression doesn't change, but he nods once barely perceptible.
36. He drops a business card on Sing's back: 'Kitchen. Tomorrow. 5 AM. Don't be late.'
37. The Bodyguard opens the door as the Master walks away without another word.
38. The crowd erupts in confused chatter as Elias and Sing remain frozen on the floor.
39. Outside in the rain, Sing takes the card from his back and stares at the card, then at Elias both soaking wet and shivering.
40. Elias grins stupidly and Sing nods slowly as they argue about who gets to hold the card.

Figure 18: **Example 5 minute (40 scenes) scenario from LVBENCH-C, Character State Evolving**: Characters appear (scenes 1-4), disappear for 10 scenes (5-14), then reappear with evolved states (15-40).

Scenes 1-5 (Lantern room - Morning routine):

1. The lantern room features crystal-clear windows and polished brass gears under a bright, cloudless morning sky.
2. The lighthouse keeper wipes a stray smudge off the massive glass lens.
3. A seagull perches on the exterior railing, visible through the lantern room glass.
4. The keeper checks his pocket watch and notes the time in a small leather logbook.
5. Sunlight reflects off the brass machinery, casting bright spots across the lantern room floor.

Scenes 6-15 (Lighthouse base - Supply delivery & storm warning):

6. Down at the lighthouse base, the heavy iron door stands open to the salty breeze.
7. A supply boat docks at the stone pier nearby, tossing ropes to the keeper's assistant.
8. The assistant carries crates of oil and food toward the lighthouse base.
9. The keeper's orange tabby cat darts into the lighthouse base, searching for shadows.
10. The assistant stacks the heavy crates against the curved stone wall of the lighthouse base.
11. Waves begin to chop more aggressively against the rocks surrounding the lighthouse base.
12. The sky turns a hazy gray as the assistant looks up from the lighthouse base entrance.
13. He begins to haul the first oil canister into the center of the lighthouse base.
14. The assistant closes the heavy iron door of the lighthouse base as the wind picks up.
15. A radio on a small table in the lighthouse base crackles with a storm warning.

Scenes 16-20 (Lantern room - Storm preparation):

16. The lantern room now has a slight layer of condensation on the glass and the sky outside is dark gray with gathering clouds.
17. The keeper pours fresh oil into the lighting mechanism of the lantern room.
18. The wind whistles through the small vents at the top of the lantern room.
19. The keeper strikes a long match, his face illuminated in the dim lantern room.
20. A first drop of rain strikes the exterior of the lantern room window.

Scenes 21-30 (Spiral staircase - Assistant's climb):

21. On the spiral staircase, the assistant slowly climbs the narrow stone steps with a flashlight.
22. The flashlight beam bounces off the damp walls of the spiral staircase.
23. He pauses to catch his breath on a small landing of the spiral staircase.
24. The sound of the assistant's boots echoes rhythmically up the spiral staircase.
25. Rainwater begins to seep through a tiny crack in the masonry of the spiral staircase.
26. The assistant continues his ascent, the spiral staircase feeling tighter and steeper.
27. He wipes sweat from his forehead as he navigates the curve of the spiral staircase.
28. The metal handrail of the spiral staircase feels cold and slippery under his grip.
29. A distant roll of thunder vibrates through the stones of the spiral staircase.
30. The assistant reaches the top hatch leading away from the spiral staircase.

Scenes 31-35 (Lantern room - Storm intensifies):

31. In the lantern room, the rotating beam is now active and rain is lashing violently against the windows in the darkening afternoon.
32. The keeper watches the beam sweep across the turbulent white-capped waves from the lantern room.
33. A sudden gust of wind makes the glass panes of the lantern room rattle in their frames.
34. The keeper adjusts the rotation speed, his shadow dancing across the lantern room walls.
35. Lightning flashes, momentarily blinding the keeper inside the lantern room.

Scenes 36-45 (Lighthouse base - Storm preparation):

36. Back at the lighthouse base, the floor is now wet with tracked-in mud and the room is lit by a single flickering electric bulb.
37. The assistant searches a cabinet in the lighthouse base for emergency flares.
38. Water begins to wash over the stone pier outside the lighthouse base.
39. The assistant secures the internal bolts on the iron door of the lighthouse base.
40. The radio in the lighthouse base is now emitting only static.
41. The orange tabby cat huddles under the supply crates in the messy lighthouse base.
42. The assistant grabs a heavy raincoat from a hook in the lighthouse base.
43. The sound of the ocean crashing against the lighthouse base becomes a deafening roar.

44. The assistant prepares a thermos of hot coffee in the dim lighthouse base.
45. He looks up at the ceiling of the lighthouse base, listening to the storm's fury above.

Scenes 46-50 (Lantern room - Electrical failure):

46. The lantern room is now engulfed in a full storm at night, with the main beam cutting through sheets of rain and the air thick with the smell of ozone.
47. The keeper struggles to keep his balance in the lantern room as the tower sways slightly.
48. A massive wave sends spray high enough to coat the lantern room windows in salt and foam.
49. Sparks suddenly fly from a control panel in the corner of the lantern room.
50. The main light flickers and dies, plunging the lantern room into darkness before emergency red lamps activate.

Scenes 51-60 (Spiral staircase - Emergency ascent):

51. The spiral staircase is now pitch black except for the assistant's swaying flashlight beam.
52. The assistant hurries up the spiral staircase, his breathing heavy and panicked.
53. A small stream of water is now flowing down the steps of the spiral staircase.
54. The assistant trips on a slick step but catches himself on the spiral staircase railing.
55. He shouts the keeper's name, his voice echoing hollowly up the spiral staircase.
56. The flashlight beam reveals the rising water at the bottom of the spiral staircase.
57. The assistant reaches the upper section of the spiral staircase, his clothes soaked through.
58. He pushes against the heavy hatch at the top of the spiral staircase with all his strength.
59. The metal of the spiral staircase groans under the pressure of the wind outside.
60. The assistant finally emerges from the spiral staircase into the upper deck.

Scenes 61-65 (Lantern room - Fighting the fire):

61. The lantern room is now filled with smoke from the shorted electrical panel and the windows are cracked, letting in freezing rain and wind.
62. The keeper uses a manual crank to keep the emergency light turning in the smoke-filled lantern room.
63. The assistant enters the lantern room and immediately grabs a fire extinguisher.
64. Together, they fight the small electrical fire in the corner of the lantern room.
65. The emergency red light reflects off the jagged cracks in the lantern room glass.

Scenes 66-75 (Lighthouse base - Storm aftermath):

66. At the lighthouse base, the iron door is partially buckled from repeated wave impacts and the floor is covered in several inches of seawater.
67. The supply crates in the lighthouse base have been knocked over by the force of the vibrations.
68. The orange tabby cat has climbed to the top of the highest shelf in the flooded lighthouse base.
69. The assistant arrives back at the lighthouse base to survey the damage.
70. He begins using a hand pump to clear the water from the lighthouse base.
71. The morning sun begins to peek through the buckled door of the lighthouse base.
72. The assistant finds the logbook floating in the puddle on the lighthouse base floor and rescues the shivering orange tabby cat from the shelf.
73. He begins to organize the debris in the ruined lighthouse base.
74. The wind has died down to a whisper around the exterior of the lighthouse base.
75. The assistant opens the door of the lighthouse base to see a calm, blue sea.

Scenes 76-80 (Lantern room - Morning after):

76. The lantern room is now calm in the bright morning light, with shattered glass on the floor and puddles of rainwater reflecting the sunrise.
77. The keeper sits on a stool in the lantern room, staring out at the horizon.
78. He begins the long process of sweeping up the glass shards in the lantern room.
79. The assistant enters the lantern room with two mugs of steaming coffee.
80. They both stand in the damaged lantern room, watching the first ship of the day pass safely by.

Figure 19: **Example 10 minute (80 scenes) scenario from LVBENCH-C, Environment State Evolving**: The lantern room, lighthouse base, and spiral staircase appear and disappear across non-consecutive scenes, with each location's state evolving naturally through a storm sequence.

H METHODOLOGY PROMPTS

H.1 MVMEM INITIALIZATION PROMPTS

H.1.1 MVMEM INITIALIZATION: REASONING, EQUATION (1)

```
You are an expert cinematographer. Deeply analyze this video story:

User Prompt:
{user_prompt}

Scenes:
{scenes}

Deep Analysis Task:
1. Determine the key characters and objects that will be shown in the video from the user
prompt. Write a concise description of how each one originally looks. Focus on only the key
characters and objects with motion.

For each character or object, provide:
- 'name': A unique identifier (e.g., "The Protagonist", "Red Silk Dress").
- 'description': A brief description alone without any other details (e.g., appearance,
clothing, colors, style) consistent with relevant backgrounds.

2. Generate the list if entities strictly as a JSON object:
```json
[{{"name": "...", "description": "..."}}]
```

3. Next, reflect back to the storyline. Are there any dynamic objects or characters in the
storyline that were missed?

4. Finally, refine the list of characters and objects with their short descriptions as a JSON
array.

Reasoning:
1. ...
2. ...
3. ...
4. ...
```json
[{{"name": "...", "description": "..."}}]
```
```

```
You are given a set of named visual references (characters, objects, environments) for a
video.
For each reference, identify what single other reference it depends on for visual consistency
during image synthesis.

References:
{json.dumps(refs, indent=2)}

Guidelines:
- Environments/backgrounds are often roots, but may depend on other environments.
- Characters typically depend on their primary environment.
- Objects typically depend on the environment or character they are associated with.
- No cycles allowed.

Return strictly as JSON list:
```json
[
  {{"name": "ref_name", "depends_on": "other_ref_name or null"}},
  ...
]
```
```

```
You are an expert spatial cinematographer. Deeply analyze this video story:

User Prompt:
{user_prompt}

Scenes:
{scenes}

Deep Analysis Task:
Step 1. First, identify what are the main environments that the movie takes place in.

Step 2. Then, for each single environment, list what objects, furniture, and elements MUST be
inside this space that facilitate all the scenes in the story smoothly. Do not combine
multiple environments into one.
```

```
Step 3. Next, reflect back to the storyline. For all scenes within/related to that
environment, is there anything missing from that environment?

Step 4. Finally, refine the list of environments with key and detailed descriptions of
necessary things only.

Step 5. Return your step-level reasonings and deep analysis in detailed text form.

Notes: Do not create unnecessary/transition environments. Stick closely to the story context
and planned scenes.

Reasoning:
1. ...
2. ...
3. ...
4. ...
5. ...

```json
[{{"environment_name": "...", "detailed_spatial_description": "description of a single
unified environment with all
necessary elements"}}]
```
```

One note here is that we require the environments to be comprehensive at the global references, as most of environments' details are static. Meanwhile, entity descriptions are minimal at this stage to facilitate state evolution; their specific details are supplemented during the frame prompting steps.

H.1.2 MVMem Initialization: Environment Synthesis

```
Generate a single unified image viewing this entire environment with all its elements
following the {env_description}. All elements must be in one cohesive scene, not split into
multiple sub-images or panels. Do not include any humans unless explicitly specified.
```

H.1.3 MVMem Initialization: Entity Synthesis

```
Generate a professional, high-fidelity, single image of {name}: {description}. A single,
centered, full image, cinematic lighting strictly on a solid, clean, all-white background. If
{name} is an object, exclude all human or animal faces unless it's explicitly required.
```

H.2 A^{2}RD Prompts

H.2.1 Prompt for Adaptive Segment Generation Mode (All Scenes), Equation (2)

We implement Equation (2) via segmenting all scenes into consecutive groups. A scene's mode is extrapolation if it is not the last in its group; otherwise, its mode is interpolation. See the prompt below.

```
Analyze the following video scenes and segment them into contiguous groups.

Scene Indices: {scene_indices}

Scenes:
{scenes}

A contiguous segment is a group of consecutive scenes that:
1. Share the same physical environment/location (e.g., interior vs exterior are different
environments).
2. Are temporally continuous (no time jump between them).
3. Have action that flows directly from one scene to the next.

Segmentation Rules (Evaluate in order):
- Priority 1 (Moving Environments): In a moving environment (characters/subjects are moving,
e.g., walking, driving, running), zoom in/out or framing changes remain continuous.
- Priority 2 (Static Environments): In a static environment (characters/subjects remain in
the same place), a significant zoom in/out or framing change (e.g., wide shot to close-up)
starts a new segment. This explicitly forces the current segment to end in a static state.
- Hard Boundaries: A new segment always starts when the environment/location changes
fundamentally, or there is a time jump. Note that interior and exterior are considered
different environments even if related (e.g., inside car vs outside car).

Reason over all the scenes one-by-one first, then return as JSON (list of lists of scene
indices):

Reasoning:...

```json
[[...], [...], ...]
```
```

```
Note: Every scene index from {scene_indices} must appear exactly once.
```

H.2.2 Prompt to Obtain $\mathcal{V}_{\text{REL}}$ (All Scenes), Equation (3)

```
Analyze the following video scenes and determine which previous scene (if any) is spatially
and temporally contiguous with each scene.

Scenes:
{scenes}

For each scene, determine if there is AT MOST ONE previous scene that is contiguous with it.
Two scenes are contiguous if:
1. They share the same physical environment/location
2. They are temporally continuous (no time jump between them)
3. The action flows directly from one to the other

If no previous scene is contiguous, return empty string "" for that scene.

Return as JSON:
```json
{{
  "0": "",
  "1": 0,
  "2": "",
  "3": 2,
  ...
}}
```

Note: Scene 0 always has empty string. Only select AT MOST ONE contiguous scene (the most
recent one if multiple exist).
```

H.2.3 Prompt to Obtain $\mathcal{S}_{\text{REL}}$ (All Scenes), Equation (3)

```
Analyze the following video scenes and determine which PREVIOUS scenes contribute to the
visual appearance of each scene.

Scenes:
{scenes}

For each scene, use chain-of-thought reasoning to identify the TOP 10 most relevant PREVIOUS
scenes (with lower indices) for visual consistency.

Step 1: Reasoning
For each scene, think through:
- What objects appear in this scene? Which previous scenes first showed these objects?
- What characters appear in this scene? Which previous scenes established these characters?
- What physical spatial environment is this scene in? Which previous scenes showed this
physical spatial arrangement?
- Which scenes are most visually important for maintaining consistency?

Step 2: Selection
Based on your reasoning, select the top 10 most relevant previous scenes for each category:
1. Objects - which previous scenes show objects that appear in this scene
2. Characters - which previous scenes show characters that appear in this scene
3. Environment - which previous scenes show the same physical spatial arrangement and layout

Return as JSON:
```json
{{
  "reasoning": {{
    "0": "Scene 0 reasoning...",
    "1": "Scene 1 reasoning...",
    ...
  }},
  "relevant_scenes": {{
    "0": {{"objects": [], "characters": [], "environment": []}},
    "1": {{"objects": [0], "characters": [0], "environment": [0]}},
    "2": {{"objects": [0, 1], "characters": [1], "environment": [0, 1]}},
    ...
  }}
}}
```

Note:
- Scene 0 always has empty lists.
- Each scene can only reference PREVIOUS scenes (lower indices).
- Each category MUST always include at most three immediately previous scenes/
- Limit to TOP 10 most relevant scenes per category, prioritizing most recent and most
visually important.
```

H.2.4 PROMPT TO RETRIEVE RELEVANT GLOBAL REFERENCES (ALL SCENES), EQUATION (4)

```
Analyze the following video scenes and determine which reference entities and environments
are relevant to each scene.

Scenes:
{scenes}

Reference Entities and Environments:
{json.dumps([f"{name}: {anchors[name]}" for name in anchor_names], indent=2)}

For each scene, identify which reference anchors (characters, objects, environments) appear
or are relevant to that scene.

Return as JSON:
```{{
  "0": ["anchor_name1", "anchor_name2"],
  "1": ["anchor_name1", "anchor_name3"],
  ...
}}```

Note:
- Only include anchors that are actually present or relevant in each scene.
- Anchor names must be selected from "{list(self.identity_anchors.keys())}".
```

H.2.5 PROMPT TO RETRIEVE THE BEST END-OF-SHOT FRAME FOR SCENE CONTINUATION, $\text{MLLM}^{\text{IMG}}_{\text{RETR}}$, EQUATION (5)

```
You are a video continuity judge. Given end-of-shot frames ({image_mapping}) from a
multi-shot video and a new scene context, determine which frame is the best starting point
for the new scene.

Original video context (contiguous scene):
{old_caption}

New scene context:
{curr_scene_description}

For each frame, analyze whether the subject, motion, and environment can naturally lead into
the new scene. Then decide which frame (if any) is the best continuation point.

Return JSON:
```json
{{"best_index": <{index_options} or null if none>, "reasoning": "..."}}
```
```

#### H.2.6 PROMPT TO GENERATE FRAME PROMPTS, $\text{MLLM}^{\text{IMG}}_{\text{PGEN}}$ IN EQUATION (5)

We implement a two-step process for the $\text{MLLM}^{\text{img}}_{\text{pgen}}$ function in Equation (5). In the first step, we generate all frame prompts based on the "All Scenes" level to capture inter-scene dependencies. In the second step, during autoregressive generation, we refine each frame prompt given the memory context to integrate relevant historical details. The prompts are provided below.

```
You are an expert Cinematographer creating visually consistent frames for a cohesive video
story. Given the scenes, environment descriptions, generate detailed image prompts for the
beginning frame of each scene and an ending frame for the final scene only.

Scenes:
{scenes}

Reference Anchor Descriptions:
{ref_frame_descriptions}

Visual Consistency Guidelines:
1. **Characters:** Maintain identical appearance (face, hair, body type, age) across all
scenes unless the story explicitly describes changes. Clothing may change if narratively
justified (e.g., changing for work, evening wear).
2. **Visual Style:** Establish consistent artistic direction, color grading, and
cinematographic approach across all frames.
3. **Environments:** Each scene's setting must align with the reference environments. Spatial
layout and static elements of recurring locations must remain identical.
4. **Camera Angles:** When continuing from a previous scene, preserve the identical camera
angle. You must explicitly say: "This scene maintains the exact camera angle as scene(s)
X...".
5. **Lighting:** If the current scene is a continuation of a previous one within the short
period of time, preserve the identical lighting conditions. Example: "The scene has the
exact same lighting condition as scene(s) Y..."
```
```

```
6. **Strict Scene Adherence:** You MUST strictly follow the Scenes provided. DO NOT change:
   - Camera shot types (wide/medium/close-up/etc) specified in the scene
   - Actions or events described in the scene
   - Objects or props mentioned in the scene
   - Lighting conditions specified in the scene
   You may ONLY add visual details that enhance the scene WITHOUT contradicting any
   specifications.
7. **First Appearance Rule:** When a character or significant object appears for the FIRST
time in the video, the frame prompt MUST include ALL explicit and implicit appearance details
identified in Step 1 below, even if they are not the primary focus of that scene. This
complete description serves as the visual reference for all subsequent scenes.
8. **Character Presence Rule:** If an established character is logically present in a scene,
explicitly describe them in the frame even if the scene description focuses on an object or
environment.

Each frame prompt must explicitly describe the following while strictly preserving all
specifications from the original scene description:
- Visual style and artistic direction.
- Natural lighting effects that reflect the scene's temporal context (morning, noon, evening,
etc).
- Character and object descriptions.
- Environment specifications matching reference descriptions.
- Camera angle details.

Step 1: Scene frame reasoning: Reasoning about the scenes one by one about entity and
environment appearances to ensure smooth transitions and strict visual consistency.

For each scene, analyze:
1. **Explicit Character/Object/Environment Appearances:** What physical attributes are
directly stated in the scenes? (e.g., "wearing a red jacket", "holding a coffee cup")
2. **Implied Appearances:** What can be logically inferred about the characters or objects
based on the subsequent scenes? (e.g., if a character "adjusts their glasses" in scene 1,
they must be wearing glasses in scene 0 (unless explicitly stated otherwise)).
3. **First Appearance Descriptions:** For the scene that a character or object or
environment's first appearance or state changes, establish detailed visual descriptions that
will serve as the reference for all subsequent scenes.

Step 2: Return a JSON dictionary with this structure. Scene indexes begin from 0:
```json
{{"frame_prompts": [{{"scene_index": 0, "begin_frame": "detailed image prompt with
consistency reasoning"}}, {{"scene_index": {final_scene_index}, "end_frame": "detailed image
prompt for final scene with consistency reasoning"}}]}}
```
Step 1 (at least 250 words):...
Step 2:...
```

```
Analyze the video states and reference images to refine the current frame prompt for physical
plausibility.

Scenes:
{relevant_scenes}

Current Scene Description (Scene {scene_index}):
{curr_scene_description}

Current Frame Prompt:
{curr_prompt}

Video States Memory:
{relevant_video_states}

Reference Images:
{image_mappings}

Analyze and reason about:

Step 1. **Scene Analysis**: Read all the provided information and analyze the following
one-by-one (at least 100 words each):
   - Is the scene contiguous with the previous scene? (Check if they share the same
   environment and have continuous action flow)
   - What is this scene trying to display according to the scene description?
   - What visual elements (objects, characters, environment) must be carried forward from
   reference images and Video States Memory for consistency?
   - What entities (characters and objects) must be carried forward from reference images and
   Video States Memory for consistency?
   - What relative positional and directional states in the Video States Memory must be
   strictly maintained in this scene?

Step 2. **Analyze Current Prompt - Camera Angle & Physical Plausibility**: Analyze the
following one-by-one (at least 100 words each):
    - Where was the camera positioned relative to the main entities in the previous scene?
```

```
    - This camera angle can be wrong. Under this camera angle, can all the relative
    positional and directional states identified in Step 1 be maintained absolutely?

Step 3. **Other Camera Angle Possibilities**: Analyze the following one-by-one (at least 100
words each):
   - If the current camera angle cannot maintain the strict positional states identitfied in
   Step 1, analyze alternative camera angles that must maintain them:
   - For each viable alternative (front view, left/right side, rear view), analyze:
     * Does this angle maintain visual continuity and smooth transition with the previous
     scene's camera position?
     * How smooth would the transition be from the previous camera angle to this angle?
     * What backgrounds and environments would be visible from this angle?
     * What entities would be visible vs hidden or obstructed?
     * What are the relative spatial relations among entities from this angle:
       - Relative entity orientations (which direction each entity faces relative to others)
       - Relative camera position (where camera is positioned relative to main subjects)
       - Relative positions between entities (distances and arrangements)
       - Depth layers (foreground/midground/background organization)

   - If the current camera angle maintains smooth continuity with previous scenes, skip this
   step and proceed to Step 4.

Step 4. **Refine the Prompt**. Discuss each of the following in at least 100 words:
   - If the scene is contiguous from the previous scene, explicitly add the instruction
   "Maintain exact camera angle and relative positional/directional states with Scene X"
   - Only change required elements that need correction for physical plausibility and
   consistency
   - Keep the rest of the original prompt unchanged to preserve the intended narrative and
   style

   - MUST specify the following spatial details in the prompt that must be consistent with
   Video States Memory:
        * Relative entity orientations (which directions the entities are facing relative to
        each other or environment features)
        * Relative camera position (camera position relative to the main subjects)
        * Relative positions between entities (with approximate distances if relevant, e.g.,
        "the woman is 5 meters to the left of the car")
        * Depth layer organization (what's in foreground/midground/background)
    - For spatial issues, avoid adding instructions such as "facing to the right/left of the
    camera" as these are often impossible to control and can cause more issues. Focus on the
    relative to the environment or other entities instead, such as "the car is backed in with
    the rear facing the door" or "the car is on the left side of the road". This must be
    specific and descriptive enough.

- Refined prompt: MUST specify what environmental entities will be visible, will NOT be
visible in the frame, and the spatial details.

Step 5. **Refine your refined prompt**:
    - Make sure all the spatial details are cleary specified and consistent with the Video
    States Memory and reference images.

Provide the reasonings for above steps, and then finally return the refined prompt in the
following JSON format:
Respond with JSON:
```
{{
  "refined_prompt": "the improved prompt"
}}
```
```

H.2.7 Prompt to Synthesize Frames, TI2I in Equation (5)

```
You are an expert Cinematographer. Given the following storyline from user:

Scenes:
{relevant_scenes}

Generate an image frame with strict visual consistency with reference images:
{final_frame_prompt}

Ensure Narrative Progression with the reference images as specified in the scenes. The
reference images are provided in order with the following mapping: {reference_image_indexes}.
```

H.2.8 Prompt to Generate Video Prompts, $\text{MLLM}_{\text{PGEN-VID}}$ in Equation (6)

```
You are a professional video producer. Generate a detailed video prompt for a
{scene_length}-second video{"that transitions from the beginning frame to the ending frame"
if end_frame_path else "starting from the beginning frame"}.

Scenes:
{relevant_scenes}
```

```
Current Scene (Scene {scene_idx}):
{current_scene}

Video States Memory (for continuity):
{relevant_video_states}

Previous Scene Video Prompt (for camera/motion continuity):
{previous_video_prompt if previous_video_prompt else "None (this is the first scene)"}

Your task is to generate a comprehensive, narrative prompt for the next video segment that
meaningfully progresses the story from the existing prompts and storyline.

Step 1. Read the scenes and understand the overall narrative arc.
Step 2. Start EXACTLY from the beginning frame (Image 1)
Step 3. {"End EXACTLY at the ending frame (Image 2). Visually inspect Image 2 and explicitly
describe its composition, entity states, and environment in the video prompt as the final
state the video must reach." if end_frame_path else "End naturally to best fulfill the
Current Scene description"}
Step 4. Maintain continuity from Previous Video States and Previous Scene Video Prompt:
   - Entity states: character appearance, clothing, accessories must remain consistent
   - Environment states: ground texture, landmarks, lighting must remain consistent
   - Motion states: ongoing actions, camera movement direction, and environmental motion
   (e.g., train vibrations, car movement, boat rocking) must continue naturally from where
   the previous scene's prompt left off
   - CRITICAL: Camera direction must NOT reverse or contradict the previous scene. If the
   previous scene's camera was moving/orbiting in a certain direction, this scene must
   continue in the same direction or come to a natural stop, never reverse.
   **Explicitly specify these continuing elements in your video prompt**
Step 5. Ensure narrative/story logical progression: the actions and events in this scene must
make sense as a continuation of what happened in the previous scenes. The characters'
activities, positions, and interactions should flow naturally from the previous scene's
ending state.
Step 6. If the scene involves multiple distinct actions or camera transitions, break the
description into temporal segments (e.g., "Scene 1: ... Scene 2: ... Scene 3: ...") to
clearly specify what happens in each part.
   - IMPORTANT: If the scene involves showing content on a phone/tablet/screen, do NOT
   describe it as "camera zooms into the phone". Instead, treat it as two separate temporal
   segments: Segment 1 describes the character holding/interacting with the device, Segment 2
   describes the screen content as if it fills the entire frame (a direct cut to the screen,
   not a zoom).

Image Mapping: [Begin Frame{", End Frame" if end_frame_path else ""}]
{'' if end_frame_path else 'IMPORTANT: There is NO end frame. Do NOT reference "Image 2" or
any end frame constraint in the video prompt.'}

In the video prompt, cover the following:
{self.SCENE_PROMPT_TEMPLATE}

Example:
Example:
A professional 8-second video featuring [the entities].
The video opens with [opening frame description]... The [entities' activities]...
{"Finally, the video concludes with [closing frame description]..." if end_frame_path else
"The video progresses naturally to fulfill the scene description..."}
[Any motions that must be continue from Previous Video States for continuity if
applicable]...
The camera [camera movement/angle description, maintaining continuity with Previous Video
States if applicable].
...

Respond with JSON:
```json
{{
  "video_reasoning": "explain your reasoning for how you structured the video prompt and
  whether temporal segments are needed",
  "video_prompt": "your detailed video prompt"
}}
```
```

H.2.9 Prompt to Synthesize Video Segments, TI2V in Equation (6)

We use the generated prompt in above step to synthesize the video segment.

H.3 HITS Self-Refinement Prompts: MLLM-Judge

H.3.1 Prompt to Extract Frame States, $\text{MLLM}^{\text{IMG}}_{\text{EXT}}$ in HITS, Section 3.3

```
Analyze this frame and extract the states of all entities.

Current Scene Description (Scene {scene_index}):
{curr_scene_description}
```

```
Frame Prompt:
{frame_prompt}

Instructions:
1. Identify every entity (characters, objects, environments) in the frame.

2. For each entity, extract:
   - type: "character", "object", or "environment"

   - identity: Features and appearance that identify this entity across scenes:
     * Physical features (body type, facial structure, distinctive marks)
     * Clothing items and colors
     * Hair style and color
     * Accessories
     * For objects: shape, design, color, material
     * For environments: architectural features, layout, defining characteristics

   - state: Current observable state that may change between scenes:
     * Position: Relative position to other entities ONLY (e.g., "on the left side of X",
     "behind/perpendicular to Y", "between X and Y"). DO NOT use frame-relative terms like
     "center", "foreground", "left side of frame".
     * Orientation: Which direction the entity faces relative to other entities or
     environment features ONLY (e.g., "facing toward the door", "facing away from X", "facing
     the same direction as Y"). DO NOT use frame-relative terms like "facing left", "facing
     the camera".
     * Depth: Relative depth to other entities (e.g., "closer to camera than X", "in front of
     Y", "behind Z")
     * Posture/arrangement (for characters: standing/sitting, limb positions; for objects:
     open/closed, arrangement)
     * Condition (clean/dirty, intact/damaged, etc.)

   Be specific with colors (exact shades) and directions. All positions and orientations must
   be relative to other entities, NOT to the frame or camera.

3. For spatial_relations, describe how entities relate spatially to each other.

Return the scene graph in this exact JSON format:
```json
{{
  "entities": {{
    "entity_name": {{
      "type": "character|object|environment",
      "identity": "features and appearance that identify this entity",
      "state": "current observable state",
    }}
  }},
  "spatial_relations": [
    {{"subject": "entity1", "relation": "relation_type", "object": "entity2"}}
  ]
}}
```
```

H.3.2 Prompt to Extract Video States, $\text{MLLM}_{\text{EXT}}^{\text{VID}}$ in HITS, Section 3.3

```
Analyze this video segment comprehensively. Provide at least 400 words total across all
sections.

Current Scene Description (Scene {scene_index}):
{curr_scene_description}

Known Entity States (from begin frame):
{entities_json}

Extract the following from the full video segment:

1. **Supplementing Missing Elements**
- Identify any NEW entities (characters, objects, environments) that appear after the begin
frame but were not in the known entity states.
- For each new entity, provide a full appearance description.

2. **Identity Changes**
- For all entities (known and new), describe any appearance details revealed throughout the
video that were not visible in the begin frame (e.g., full body, accessories, clothing
details, object condition).
- For characters: focus on hair, clothing, physical appearance.
- For objects: focus on color, shape, condition, visible attributes.
- For environments: focus on lighting, weather, background elements, and spatial
arrangements.

3. **Motions**
```

```
- For each element (known and new), describe its motion using the template: "At the
beginning, [...]. In the middle, [...]. At the end, [...]."

4. **Camera Dynamics**
- Describe camera movement, angle changes, zoom, and panning throughout the video using the
same temporal template.

Return as JSON:
```json
{{
 "new_entities": {{
   "<entity_name>": "full appearance description"
 }},
 "identity_changes": {{
{entities_template}
 }},
 "motions": {{
{entities_template}
 }},
 "camera": "At the beginning, [...]. In the middle, [...]. At the end, [...].",
}}
```
```

H.3.3 Prompt to Judge Frame - Consistency over Images (Entity, Environment, Narrative), Equation (7)

```
Evaluate the generated frame against reference images for consistency and progression.

Scenes:
{relevant_scenes}

Current Scene (Scene {scene_index}):
{scenes[scene_index]}

Frame Prompt:
{frame_prompt}

Image Mapping:
{image_mapping}

What's wrong with the current image compared to its references with mappings specified in
Image Mapping?

3. Entity Reference Consistency (1-10): First, identify which reference images (by Image
number and label) show the same characters or objects that appear in the current frame. Then,
using only those relevant references, check if any entity violates their appearance. If there
are visual conflicts (e.g., object appearance changes, character look changes,
clothing/design changes, distinctive feature changes) that are NOT explicitly described in
the Frame Prompt, this is a violation. Compare each character and object one-by-one:
   - Focusing on every characters (emphasizing appearance, cloths, hair, accessories, etc)
   and objects (emphasizing furniture, devices, etc) visible in the frame. For each one,
   identify which reference images show that same entity, then check the EXACT appearance,
   colors, clothing/design, distinctive features against those references
   - State what matches and what differs from the reference images
Be very critical and detailed. Describe in detail in at least 200 words.

4. Environment Reference Consistency (1-10): First, identify which reference images (by Image
number and label) show the same environment/location as the current frame. Then, using only
those relevant references, check if the environment violates its appearance. If there are
spatial conflicts (e.g., position inconsistencies, spatial arrangement changes, architectural
changes) that are NOT explicitly described in the Frame Prompt, this is a violation:
   - Focusing on every spatial layout, architectural elements, lighting, overall setting
   - State what matches and what differs from the reference images
Be very critical and detailed. Describe in detail in at least 200 words.

5. Narrative Progression (1-10): What's wrong with the logical progression of the current
frame compared to its prior frame regarding the entities, the environment, and logical
lighting? The frame must NOT be identical or nearly identical to previous reference frames
unless the Frame Prompt explicitly describes a static or unchanged scene. There must be
visible change in entity positions/actions, environmental elements, or lighting conditions
that aligns with the storyline progression. Lighting must progress consistently with time: if
the previous frame shows late afternoon, the current frame must be late or even later unless
the story explicitly involves a time jump. If the frame looks like a duplicate or shows no
meaningful progression in entities or environment when the Frame Prompt indicates change
should occur, this is a violation. Discuss in details in at least 250 words.

Respond with JSON:
```json
{{
  "entity_reference_consistency": <score 1-10>,
  "environment_reference_consistency": <score 1-10>,
```
```

```
  "narrative_progression": <score 1-10>,
  "reasoning": "detailed explanation of consistency and progression issues"
}}
```

H.3.4 Prompt to Judge Frame - Consistency over Images (Spatial Logicalness), Equation (7)

```
Carefully examine the spatial arrangements and object positions across these images.

Scenes:
{relevant_scenes}

Current Scene (Scene {scene_index}):
{scenes[scene_index]}

Frame Prompt:
{frame_prompt}

Image Mapping:
{image_mapping}

Note: Images labeled "anchor_*" show the canonical spatial layout and object arrangements.
The current frame MUST match these anchor layouts unless the Frame Prompt explicitly
describes changes.

What's wrong with the current image compared to existing images regarding the spatial
environment? Reasoning in at least 250 words.

Respond with JSON:
```json
{{
  "spatial_logicalness": <score 1-10, where 10 means nothing wrong>,
  "reasoning": "detailed analysis of spatial issues or 'nothing wrong'"
}}
```
```

H.3.5 Prompt to Judge Frame - Textual States, Equation (7)

```
Compare the extracted states with relevant previous scenes and the storyline to identify
discrepancies.

Scenes:
{relevant_scenes}

Current Scene (Scene {scene_index}):
{scenes[scene_index]}

Video States Memory:
{relevant_video_states}

Current Scene States:
{video_states}

Instructions:
Reasoning in at least 250 words to critically analyze consistency smartly:

Step 1: **Analyze Objects**
- What objects appear in the current scene's extracted states?
- How do these objects compare to their states in relevant previous scenes?
- What discrepancies exist (if any)? Note that if new objects appear in this scene that
weren't in earlier scenes, and there's no narrative storyline for their appearance, this
counts as a discrepancy.
- Are these discrepancies justified by the current scene description?
- Note that environmental changes due to natural video progression (e.g., spatial
relationships changing as character moves, new buildings coming into view) should NOT be
counted as discrepancies. Only flag static attributes that must be consistent.

Step 2: **Analyze Characters**
- What characters appear in the current scene's extracted states?
- How do these characters compare to their states in relevant previous scenes?
- What discrepancies exist (if any)? Note that if new characters appear in this scene that
weren't in earlier scenes, and there's no narrative storyline for their appearance, this
counts as a discrepancy.
- Are these discrepancies justified by the current scene description?

Step 3: **Analyze Environment**
- What physical spatial environment appears in the current scene's extracted states?
- How does this physical spatial environment compare to its state in relevant previous
scenes?
```

```
- Strictly spot any spatial discrepancies or illogical environmental arrangements exited (if
any). Note that if the physical spatial environment remains the same but new lights,
atmosphere, or other environmental elements appear in this scene that weren't in earlier
scenes, and there's no narrative storyline for their appearance, this counts as a
discrepancy.
- Are these discrepancies justified by the current scene description? Normally, most of the
spatial discrepancies are not justified unless explicitly described in the scene.

Step 1 (at least 200 words):...
Step 2 (at least 200 words):...
Step 3 (at least 200 words):...

Return as JSON:
```json
{{
  "entity_objects": "detailed reasoning about objects consistency...",
  "entity_object_state_score": <score 1-10>,
  "entity_characters": "detailed reasoning about characters consistency...",
  "entity_character_state_score": <score 1-10>,
  "environment": "detailed reasoning about environment consistency...",
  "environment_state_score": <score 1-10>
}}
```
```

H.3.6 Prompt to Judge Frame - Basic Quality, Equation (7)

```
Evaluate if the generated frame successfully matches the following prompt and meets quality
standards:

Frame Prompt:
{frame_prompt}

Current Frame States:
{video_states}

Evaluate and score each criterion on a scale of 1-10:

1. Instruction Following (1-10): Examine every single detail in the prompt. Does the frame
faithfully capture every critical component specified in the prompt, especially lighting
conditions and environmental details? Discuss in details.
2. Physical Plausibility (1-10): Is the frame physically realistic? All objects, characters,
and environments must appear physically plausible and follow real-world physical laws unless
the frame prompt specifies otherwise. The frame must represent one unified moment in space
and time. NO "cinematic license" or artistic adjustments should excuse physical
impossibilities. NO split screens, panels, or multiple disconnected scenes. NO collage-style
layouts showing different moments or locations. Discuss in details.

Respond with JSON:
```json
{{
  "instruction_following": <score 1-10>,
  "physical_plausibility": <score 1-10>,
  "reasoning": "explanation of any issues found in detail"
}}
```
```

H.3.7 Prompt to Judge Video - Inter Consistency, Equation (9)

```
Analyze event consistency between two contiguous scenes.

Scenes:
{relevant_scenes}

Previous Contiguous Scene (Scene {previous_scene_idx}):
{previous_scene}

Current Scene (Scene {current_scene_idx}):
{scenes[scene_index]}

Previous Contiguous Scene Video States:
{pre_contiguous_video_states}

Current Scene Video States:
{video_states}

Instructions:
These two scenes are spatially and temporally contiguous (same environment, continuous action
flow). Compare the states of the current video versus previous video states to spot any
inconsistencies regarding the states, motions, and camera:
```

```
Step 1: **Analyze Entity Appearance Consistency (at least 100 words)**
- For each entity in the current scene, check if it's new or appeared in the previous scene
- Are new entities properly justified by the scene description?
- For entities appearing in both scenes, do their visual appearance and states remain
consistent?

Step 2: **Analyze Entity Motion Consistency (at least 100 words)**
- For entities appearing in both scenes, compare their motions at the end of Scene
{previous_scene_idx} with the beginning of Scene {current_scene_idx}
- Do the motions form a smooth transition regarding activities and directions?
- Identify any abrupt changes in entity motions that aren't justified by the scene
descriptions

Step 3: **Analyze Environment Appearance Consistency (at least 100 words)**
- Compare the environment layout and lighting at the end of Scene {previous_scene_idx} with
the beginning of Scene {current_scene_idx}
- Does the spatial layout (room structure, architectural elements) remain consistent?
- Does the lighting (intensity, direction, color temperature) remain consistent?
- Note: Natural environmental changes due to video progression are acceptable (e.g.,
different parts of the same location becoming visible as characters move, background scenery
shifting due to camera/character movement, gradual lighting changes)
- Only flag SIGNIFICANT inconsistencies: sudden teleportation to different locations, drastic
lighting jumps without narrative justification, or architectural impossibilities
- Since these scenes are contiguous, major environment changes should be justified by the
scene descriptions

Step 4: **Analyze Camera Consistency (at least 200 words)**
- Compare camera movement at the end of Scene {previous_scene_idx} with the beginning of
Scene {current_scene_idx}
- Since these scenes are contiguous, the camera should maintain continuity - do they form a
consistent flow?
- Identify any jarring camera transitions that break visual continuity.

Return as JSON:
```json
{{
  "entity_consistency_reasoning": "detailed analysis of entity appearance consistency",
  "entity_consistency_score": <score 1-10>,
  "entity_motion_reasoning": "detailed analysis of entity motion consistency",
  "entity_motion_score": <score 1-10>,
  "environment_consistency_reasoning": "detailed analysis of environment appearance
  consistency",
  "environment_consistency_score": <score 1-10>,
  "camera_reasoning": "detailed analysis of camera consistency",
  "camera_score": <score 1-10>
}}
```
```

H.3.8 Prompt to Judge Video - Intra Consistency and Basic Video Quality, Equation (9)

```
Evaluate the generated video against the scene description.

Current Scene:
{scenes[scene_index]}

Current Video States:
{video_states}

Grading Guidelines (apply to all criteria):
- 1-3: Major issues with main entities/objects appearance (e.g. wrong character, wrong
outfit, object replaced, impossible physics, jarring cut, different hairstyle)
- 4-6: Noticeable but minor issues with main entities/objects appearance (e.g. slight color
shade difference, minor texture variation, small accessory missing)
- 7-10: Main entities/objects appearance is perfect; only background or negligible details
may differ

Step 1: Understand the video: Watch the video carefully and read the 'Current Video State' to
fully understand what happened in the video - what entities are present, how they look, how
they move, and how the camera behaves.

Step 2: Judge against the scene description. Critically flag any error at each criterion on a
scale of 1-10. For each criterion, provide at least 150 words of detailed analysis:

1. Instruction Following (1-10): Watch the video. Does it match the scene description? Check
character appearance, entity appearance, character motions, entity motions, background and
camera against the scene description as ground truth.

2. Physical Plausibility (1-10): Watch the video. Is it physically realistic? Check character
physics, entity physics, visual artifacts, and lighting consistency purely from what you
observe in the video.
```

```
3. Narrative Progression (1-10): Watch the video. Does it flow smoothly and continuously as a
single professional shot?

4. Frame Fit (1-10): Watch the video. Does the story logically end with the provided end
frame? The end frame is the last image in the prompt, describe it in detail, then judge
whether the video leads naturally to it.

5. Character Consistency (1-10): Do NOT watch the video. Compare `identity` vs
`identity_changes` in the 'Current Video State' for each character. Any mismatch not
explicitly justified by the scene description is an inconsistency. Apply the grading
guidelines based on severity of the mismatch.

6. Object Consistency (1-10): Do NOT watch the video. Compare `identity` vs
`identity_changes` in the 'Current Video State' for each object. Any mismatch not explicitly
justified by the scene description is an inconsistency. Apply the grading guidelines based on
severity of the mismatch.

7. Environment Consistency (1-10): Do NOT watch the video. Compare `identity` vs
`environment_changes` in the 'Current Video State' for each environment entity. Any mismatch
not explicitly justified by the scene description is an inconsistency. Apply the grading
guidelines based on severity of the mismatch.

Respond with the following format:

Step 1: Understand the video: Describe what you observe in the video and the 'Current Video
State'...
Step 2: Judge each criterion with detailed analysis (at least 150 words each)...

Return the final scores in JSON format:
```json
{{
  "instruction_following": <score 1-10>,
  "physical_plausibility": <score 1-10>,
  "logical_progression": <score 1-10>,
  "frame_fit": <score 1-10>,
  "entity_character_consistency": <score 1-10>,
  "entity_object_consistency": <score 1-10>,
  "environment_consistency": <score 1-10>
}}
```
```

H.4 MAPO PROMPTS

H.4.1 PROMPT TO REASON OVER FEEDBACK (FRAME PROMPT)

```
You are analyzing an image generation prompt that failed validation.

Scene Description (MUST be fully satisfied):
{scene_description}

Original Prompt: {current_prompt}

Validation Scores: {scores}

Validation Feedback: {feedback}

Tasks:
1. READ AND UNDERSTAND the scene description. This is the ground truth that MUST be fully
satisfied.

2. READ AND UNDERSTAND the original prompt in detail. What is it trying to achieve? What are
ALL the key elements?

3. READ AND UNDERSTAND the validation feedback in detail. What specific issues were
identified? Minor issues that that human can't see or barely notice can be disregarded.

4. ANALYZE DISCREPANCIES BETWEEN PROMPT AND GENERATION:
   - Where did the generation fail? Which parts of the prompt were missed or misinterpreted?
   - List specific elements described in the prompt that were not correctly generated
   - Are we drifting away from the scene description?

5. PROMPT REASONING AND REFINEMENT (at least 250-1000 words): This step is all about prompt
reasoning and we must not discuss anything about the generation.

    Structure your analysis as follows:

    **Problematic phrases/parts:** For each issue discovered above, identify which phrases or
    instructions are causing the issues. Are there any ambiguities or contradictions in the
    prompt that could lead to misinterpretation?
```

```
    **What to add/clarify:** What specific clarifications or conflicts would resolve the
    ambiguities and help the model understand the intent better?

    **What to remove/simplify:** Are there overly specific constraints, redundant details, or
    non-essential elements that should be removed or simplified?

Example format:
    - **Problematic:** "The scientist holds a vial with a precise two-finger grip" - This is
    overly specific and impossible for the model to control exactly...
    - **Add/Clarify:** Specify that "the scientist carefully holds a small vial" to convey
    the careful handling without impossible constraints...
    - **Remove/Simplify:** Remove the detailed finger position requirements ("only distal
    pads of thumb and index finger") as these add complexity without improving the core
    scene...

Notes:
    - This step 5 reasoning is critical: analyze every detail in the prompt that could cause
    the identified issues.
    - Focus entirely on prompt content and structure - do not mention generation quality or
    results in this step.
    - Pay special attention to the Scene Description and Physical Plausibility of the
    prompt's instructions.
    - Use the world "could cause" rather than "is causing" to avoid making assumptions about
    the generation - we are only analyzing the prompt here, not the generation results.


6. SPECIAL NOTES FOR SPATIAL/ENVIRONMENTAL INCONSISTENCIES:
    If the feedback mentions entity location or direction inconsistencies (e.g., "mannequin
    disappeared", "machine moved to wrong side", "table is now behind instead of left"), add
    explicit positional constraints to the refined prompt such as "[entity] remains at
    [location]" (e.g., "in the background", "on the table") or "[entity] stays [direction] of
    [reference]" (e.g., "left of the window", "behind A").
    Note: Avoid adding instructions such as facing to the right/left as these are often
    impossible to control and can cause more issues. Focus on the relative to the environment
    or other entities instead, such as "the car is backed in with the rear facing the door"
    or "the car is on the left side of the road". This must be specific and descriptive
    enough.

7. SPECIAL NOTES FOR CAMERA:
    Many times, the errors indicated in the feedback come from an impossible camera angle. In
    such cases, use a camera angle from the reference frames instead of the current prompt's
    camera specification.

8. DETERMINE MODE:
    Based on the errors identified, determine if the issues can be fixed with minor edits or
    require regeneration:
    - "edit": The errors are minor and can be fixed, such as entity states, small adjustments
    to positioning, or clarifications
    - "regenerate": The errors are major, such as fundamental scene misinterpretation,
    spatial or environmental or camera issues, missing key elements, or severe physical
    plausibility issues

Provide your step-by-step reasoning, at least 500 words:
Step 1: ...
Step 2: ...
Step 3: ...
Step 4: ...
Step 5: ...
Step 6: ...
Step 7: ...
Step 8: ...

Then return as JSON:
```json
{{
    "prompt_reasoning": "Copy exactly your Step 5 reasoning here.",
    "mode": "edit or regenerate",
    "edit_prompt": "if mode is edit, provide a one-sentence edit instruction with 'Edit the
    last reference image via...' (e.g., 'Edit the last reference image via adding the tools
    from Scene X on the table'). Otherwise, leave empty."
}}
```
```

H.4.2 PROMPT TO REASON OVER FEEDBACK (VIDEO PROMPT)

```
You are analyzing a video generation prompt that failed validation.

Current Scene (MUST be fully satisfied):
{scenes[scene_index]}

Original Prompt: {current_prompt}
```

```
Validation Scores: {scores}

Validation Feedback: {feedback}

Tasks:
Step 1. READ AND UNDERSTAND the scene description. This is the ground truth that MUST be
fully satisfied.

Step 2. READ AND UNDERSTAND the original prompt in detail. What is it trying to achieve? What
are ALL the key elements and actions?

Step 3. READ AND UNDERSTAND the validation feedback in detail. What specific issues were
identified?

Step 4. ANALYZE DISCREPANCIES BETWEEN PROMPT AND GENERATION:
   - Character identities: Did any character's appearance (hair, clothing, accessories,
   physical attributes) change unexpectedly or differ from what the scene description
   requires? Check ALL characters.
   - Object identities: Did any object's appearance (color, shape, condition, texture) change
   unexpectedly or differ from what the scene description requires? Check ALL objects.
   - Environment: Did the environment layout, lighting, or background change unexpectedly
   (sudden lighting shifts, background elements appearing/disappearing, layout changes) in
   ways not justified by the scene description?
   - Motions/directions: Which actions, movements, or transitions were missed or
   misinterpreted?

Step 5. PROMPT REASONING AND REFINEMENT (at least 250-1000 words): This step is all about
prompt reasoning and we must not discuss anything about the generation.

    Structure your analysis as follows:

    **Problematic phrases/parts:** For each issue discovered above, identify which phrases or
    instructions are causing the issues. Are there any ambiguities or contradictions in the
    prompt that could lead to misinterpretation? For video prompts, pay special attention to:
    - Character identities: Does the prompt explicitly enforce that all characters maintain
    their appearance (hair, clothing, accessories) throughout the entire video?
    - Object identities: Does the prompt explicitly enforce that all objects maintain their
    appearance (color, shape, condition) throughout the entire video?
    - Environment: Does the prompt explicitly enforce that the environment layout, lighting,
    and background remain stable and consistent throughout the entire video?
    - Motions/directions: Are action descriptions clear, sequential, and free of impossible
    constraints or hyper-specific timing?

    **What to add/clarify:** What specific clarifications would resolve the identified
    issues? For video prompts, consider:
    - Character identities: If any character's appearance changed unexpectedly, explicitly
    add constraints such as "throughout the entire video, [character] must maintain
    [appearance]"
    - Object identities: If any object's appearance changed unexpectedly, explicitly add
    constraints such as "throughout the entire video, [object] must remain [appearance]"
    - Environment: If the environment layout or lighting changed unexpectedly, explicitly add
    constraints such as "throughout the entire video, the environment/lighting must remain
    [description]"
    - Motions/directions: Add motion flow, action sequencing, and transition cues to resolve
    ambiguities

    **What to remove/simplify:** Are there overly specific constraints, redundant details, or
    non-essential elements that should be removed or simplified? For video prompts, consider:
    - Character/object descriptions that are overly detailed and could cause the model to
    hallucinate inconsistencies
    - Overly specific motion constraints that are impossible to achieve
    - Redundant or conflicting descriptions
    - Technical specifications that could be replaced with simpler visual language

Notes:
    - This step 5 reasoning is critical: analyze every detail in the prompt that could cause
    the identified issues.
    - Focus entirely on prompt content and structure | do not mention generation quality or
    results in this step.
    - Prioritize character identity, object identity, and environment consistency issues
    before motion issues.
    - Use "could cause" rather than "is causing" | we are only analyzing the prompt, not the
    generation results.

Provide your step-by-step reasoning, at least 500 words:
Step 1: ...
Step 2: ...
Step 3: ...
Step 4: ...
Step 5: ...

Then return as JSON:
```

```
```json
{{
    "prompt_reasoning": "Copy exactly your Step 5 reasoning here."
}}
```
```

H.4.3 PROMPT TO EXTRACT CONTRASTIVE PATTERNS (FRAME PROMPT)

```
Analyze successful and failed frame generation prompt refinements to extract actionable
patterns.

Successful Refinements (what worked):
{positive_samples}

Failed Refinements (what didn't work):
{negative_samples}

By comparing the original vs refined prompts, identify what changes led to success or
failure.

Extract 8-12 actionable patterns. Each pattern must:
1. Be specific and concrete (not vague advice)
2. Include a brief example (e.g., "use 'X' instead of 'Y'")
3. Be 1-2 sentences maximum

Focus on:
- **Structural changes**: How prompt organization affects results
- **Specificity**: When adding/removing detail improves generation
- **Terminology**: Which words/phrases work vs fail
- **Spatial/visual logic**: How to describe positions, angles, relationships
- **Common failures**: What consistently breaks generation

Respond with JSON:
{{
  "patterns": [
    "Pattern with brief example",
    ...
  ]
}}
```

H.4.4 PROMPT TO EXTRACT CONTRASTIVE PATTERNS (VIDEO PROMPT)

```
Analyze successful and failed video generation prompt refinements to extract key patterns.

Successful Refinements (what worked):
{positive_samples}

Failed Refinements (what didn't work):
{negative_samples}

Extract actionable patterns across these categories:

1. **Structural Patterns**: Optimal prompt organization for video, length, ordering
2. **Temporal/Motion Descriptions**: Effective ways to describe motion flow, action
sequencing, timing
3. **Continuity Instructions**: How to ensure smooth transitions and consistent framing
4. **Specificity vs. Abstraction**: When to be precise vs. general for motion/timing
5. **Conflict Resolution**: Common contradictions in action descriptions, impossible
constraints
6. **Domain-Specific Vocabulary**: Effective motion/temporal terms vs. confusing terms
7. **Negative Patterns**: Instructions that consistently fail for video, over-specifications

Return each pattern as a separate item in a list. Each pattern should be a concise,
actionable guideline (1-2 sentences).

Respond with JSON:
{{
  "patterns": [
    "Pattern 1 guideline here",
    "Pattern 2 guideline here",
    ...
  ]
}}
```

H.4.5 PROMPT TO REFINE FRAME PROMPT

```
You are refining an image generation prompt based on analysis.

Current Scene (MUST be fully satisfied):
{scenes[scene_index]}
```

```
Original Prompt: {current_prompt}

Prompt Reasoning (Analysis):
{prompt_reasoning}

Learned Prompting Patterns (apply relevant ones):
{learned_patterns_from_successful_cases}

Refine the prompt targettedly strictly following the guidelines below:
- If the analysis indicates no issues or the prompt is already optimal, return the original
prompt unchanged
- Otherwise, modify the specific parts via additions/clarifications and
removals/simplifications identified as problematic
- Apply relevant learned patterns to improve the prompt
- Keep everything else from the original prompt unchanged
- For spatial issues, avoid adding instructions such as "facing to the right/left of the
camera" as these are often impossible to control and can cause more issues. Focus on the
relative to the environment or other entities instead, such as "the car is backed in with the
rear facing the door" or "the car is on the left side of the road". This must be specific and
descriptive enough.

Return as JSON:
```json
{{
    "refined_prompt": "Your refined prompt. Make only the specific changes identified in the
    analysis - strictly keep everything else from the original prompt unchanged."
}}
```
```

H.4.6 Prompt to Refine Video Prompt

```
You are refining a video generation prompt based on analysis.

Current Scene (MUST be fully satisfied):
{scenes[scene_index]}

Original Prompt: {current_prompt}

Prompt Reasoning (Analysis):
{prompt_reasoning}

Learned Prompting Patterns (apply relevant ones):
{learned_patterns_from_successful_cases}

Refine the prompt targettedly strictly following the guidelines below:
- If the analysis indicates no issues or the prompt is already optimal, return the original
prompt unchanged
- Otherwise, modify the specific parts via additions/clarifications and
removals/simplifications identified as problematic
- Apply relevant learned patterns to improve the prompt
- Keep everything else from the original prompt unchanged

- CRITICAL: Do not change the beginning state of any entity! Their initial position, motion,
and condition at the start of the video might be inherited from the previous scene and must
be preserved exactly as in the original prompt. For example, if the car is moving at the
beginning in the Original Prompt, you must keep the car moving instead of the car is
accelerating.

- CRITICAL: By default, use a static camera. Do NOT use zoom, pan, tilt, dolly, crane, or
orbital/circling movements unless the scene explicitly requires it. If camera changes are
needed, they must be split into separate temporal segments with clear time ranges (e.g.,
"0-4s: static wide shot... 4-8s: tracking shot following subject...")

Return as JSON:
```json
{{
    "refined_prompt": "Your refined prompt. Make only the specific additions/removals
    identified in the analysis - strictly keep everything else from the original prompt
    unchanged."
}}
```
```

H.5 Section D.3's Prompts

H.5.1 Prompt to Verify Character Consistency Drift, Equation (11)

```
You are evaluating CHARACTER consistency in a long video generation.

The provided images are labeled as follows:
{image_labels}
```

```
Scene descriptions:
{past_scenes}

Current scene description (Scene {scene_idx}):
{current_scene}

Follow these steps:

Step 1: Identify visual changes: Compare the character in the reference frames to the
character in the current scene frame. Focus on the character's face, hair, identity, body
type, and overall appearance.

Step 2: Classify changes: For each difference, determine whether it is expected (justified by
the story) or unexplained.

Step 3: Evaluate: Only flag as violated (0) for CLEAR appearance changes: a completely
different person appears, the character's face/body type is obviously different, or the
character's species changes. Do NOT flag minor variations like slight outfit dirt,
accessories, lighting differences, or pose changes.

Output ONLY this JSON:
```json
{{
  "step1_visual_changes": "<list character differences>",
  "step2_classification": "<expected or unexplained for each>",
  "character_consistent": <1 if consistent, 0 if violated>,
  "reasoning": "<brief reason>"
}}
```
```

H.5.2 Prompt to Verify Object Consistency Drift, Equation (11)

```
You are evaluating OBJECT consistency in a long video generation.

The provided images are labeled as follows:
{image_labels}

Scene descriptions:
{past_scenes}

Current scene description (Scene {scene_idx}):
{current_scene}

Follow these steps:

Step 1: Identify visual changes: Compare the key objects in the reference frames to the
current scene frame. Focus on the object's identity, shape, color, and size.

Step 2: Classify changes: For each difference, determine whether it is expected (justified by
the story) or unexplained.

Step 3: Evaluate: Only flag as violated (0) for CLEAR appearance changes to key objects:
object is a completely different item, shape is obviously wrong, or color is drastically
different. Do NOT flag minor position changes, slight wear/damage, lighting differences, or
partial occlusion.

Output ONLY this JSON:
```json
{{
  "step1_visual_changes": "<list object differences>",
  "step2_classification": "<expected or unexplained for each>",
  "object_consistent": <1 if consistent, 0 if violated>,
  "reasoning": "<brief reason>"
}}
```
```

H.5.3 Prompt to Verify Environment Consistency Drift, Equation (11)

```
You are evaluating BACKGROUND consistency in a long video generation.

The provided images are labeled as follows:
{image_labels}

Scene descriptions:
{past_scenes}

Current scene description (Scene {scene_idx}):
{current_scene}

Follow these steps:
```

```
Step 1: Identify visual changes: Compare the background/environment in the reference frames
to the current scene frame. Pay close attention to:
- Architectural structure: walls, pillars, arches, ceiling, floor layout
- Spatial arrangement: relative positions of objects, furniture, doorways, ledges

Step 2: Classify changes: For each difference, determine whether it is expected (justified by
the story) or unexplained.

Step 3: Evaluate: The background must be the SAME physical space as the reference frames, not
just a similar-looking environment. Flag as violated (0) only if the room/location is clearly
different: different architecture, different layout, or a different place entirely. Do NOT
flag camera angle differences, story-driven events, or minor lighting changes.

Output ONLY this JSON:
```json
{{
  "step1_visual_changes": "<list background differences>",
  "step2_classification": "<expected or unexplained for each>",
  "background_consistent": <1 if consistent, 0 if violated>,
  "reasoning": "<brief reason>"
}}
```
```

I Full Storyboard Examples

I.1 A^{2}RD Examples

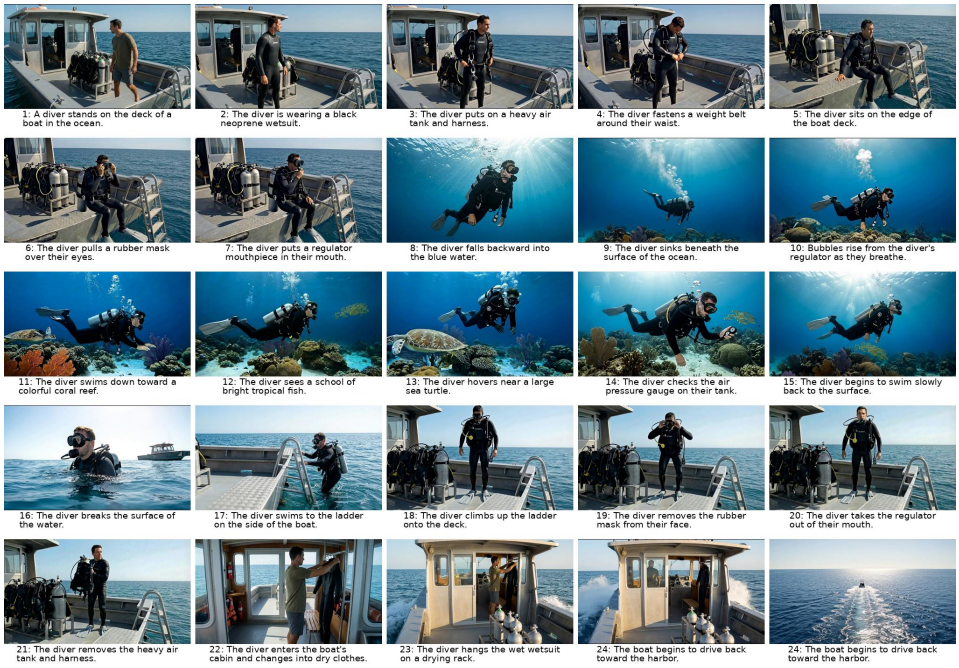


Figure 20: Storyboard synthesized by A^{2}RD of a 3-minute video from LVBench-C: *The Scuba Diver's Reef Exploration* (24 scenes).

I.2 Baseline Examples



Figure 21: Storyboard synthesized by A^{2}RD of a 5-minute video from LVBENCH-C: *The Stage Fright (Clara)* (40 scenes).

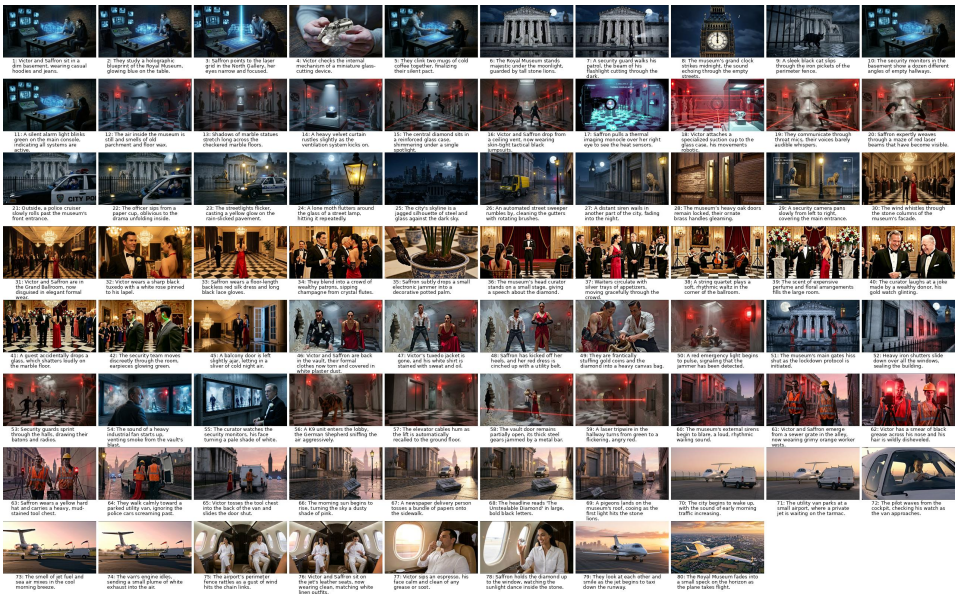


Figure 22: Storyboard synthesized by A^{2}RD of a 10-minute video from LVBENCH-C: *The Great Museum Heist* (79 scenes).

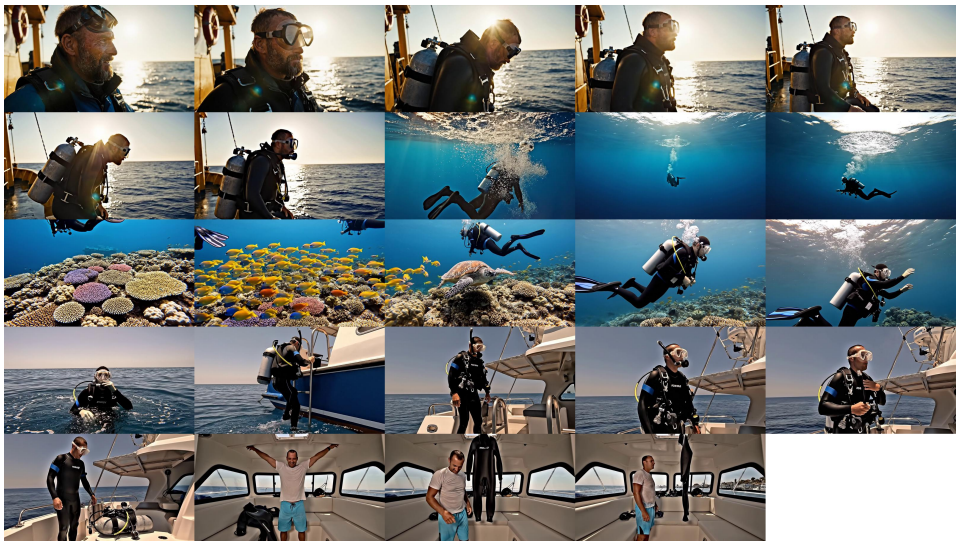


Figure 23: Storyboard synthesized by Naive-AR of a 3-minute video from LVBENCH-C: *The Scuba Diver's Reef Exploration* (24 scenes). Frequent inconsistencies in characters, objects, and environments accumulate throughout the video, severely degrading narrative coherence and visual continuity.



Figure 24: Storyboard synthesized by Naive-Par of a 3-minute video from LVBENCH-C: *The Scuba Diver's Reef Exploration* (24 scenes). Frequent inconsistencies in characters, objects, and environments accumulate throughout the video, severely degrading narrative coherence and visual continuity.

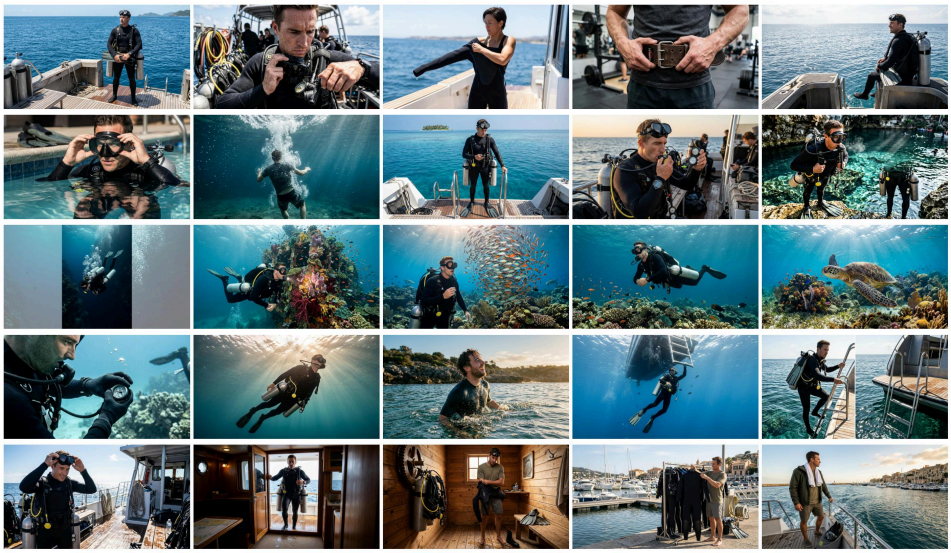


Figure 25: Storyboard synthesized by MovieAgent of a 3-minute video from LVBENCH-C: *The Scuba Diver's Reef Exploration* (24 scenes). Frequent inconsistencies in characters, objects, and environments accumulate throughout the video, severely degrading narrative coherence and visual continuity.

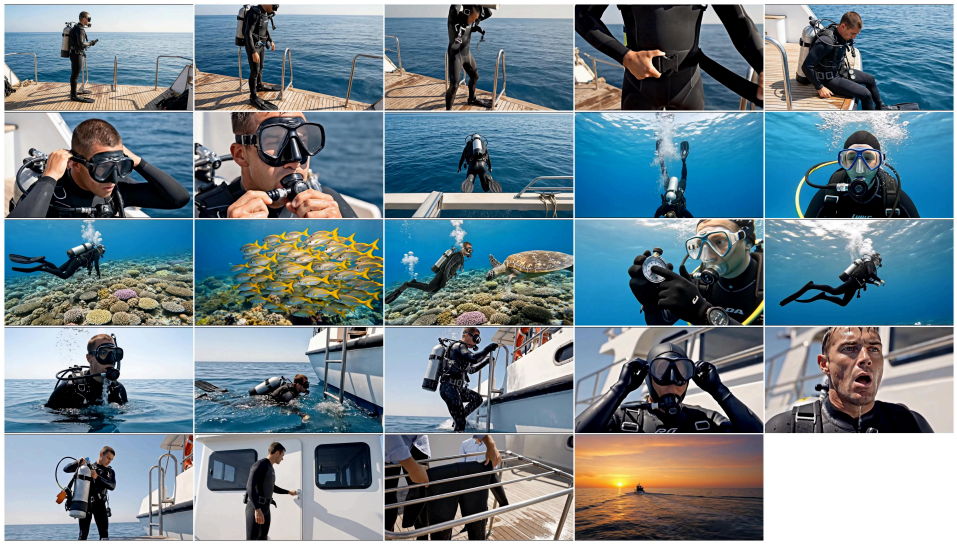


Figure 26: Storyboard synthesized by ViMax of a 3-minute video from LVBENCH-C: *The Scuba Diver's Reef Exploration* (24 scenes). Frequent inconsistencies in characters, objects, and environments accumulate throughout the video, severely degrading narrative coherence and visual continuity.

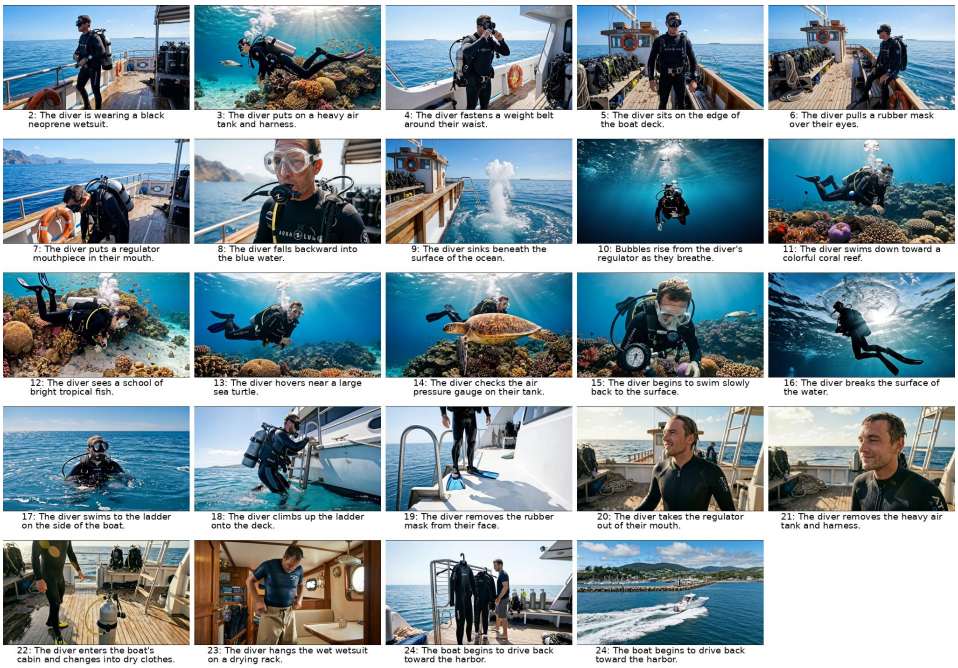


Figure 27: Storyboard synthesized by VideoMemory of a 3-minute video from LVBENCH-C: *The Scuba Diver's Reef Exploration* (24 scenes). Frequent inconsistencies in characters, objects, and environments accumulate throughout the video, severely degrading narrative coherence and visual continuity.

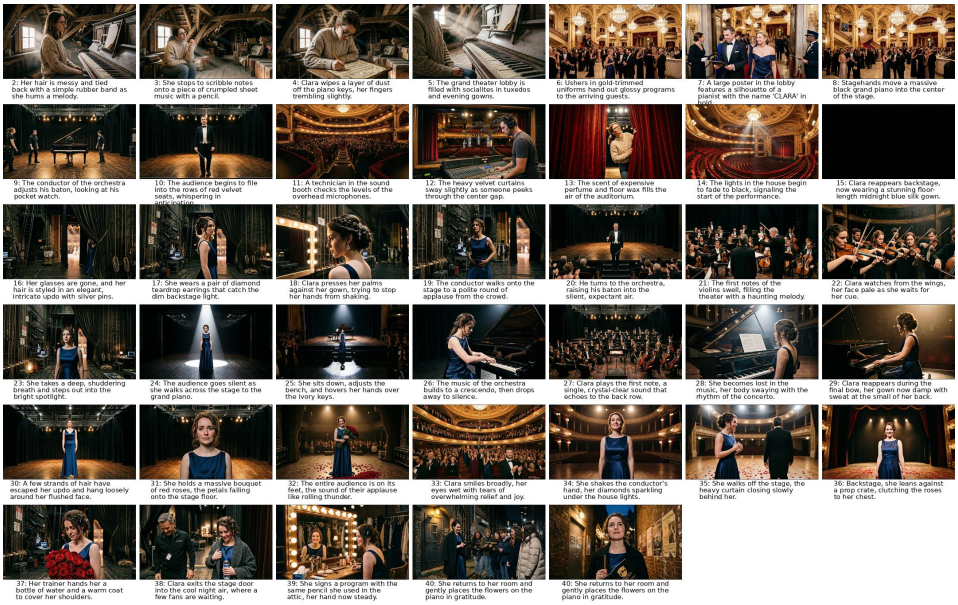


Figure 28: Storyboard synthesized by VideoMemory of a 5-minute video from LVBENCH-C: *The Stage Fright (Clara)* (40 scenes). Frequent inconsistencies in characters, objects, and environments accumulate throughout the video, severely degrading narrative coherence and visual continuity.